

UNIVERSIDAD POLITÉCNICA DE MADRID

Escuela Técnica Superior de Ingeniería y Sistemas de
Telecomunicación



PROYECTO DE FIN DE CARRERA

**Desarrollo de un sistema automático de medida para el control del consumo en
sistemas empotrados.**

JOSÉ FERNÁNDEZ GIL

SEPTIEMBRE 2014



PROYECTO FIN DE CARRERA PLAN 2000

E.T.S.I.S. TELECOMUNICACIÓN

TEMA	Optimización de energía		
TITULO	Desarrollo de un sistema automático de medida para el control del consumo en sistemas empotrados.		
AUTOR	José Fernández Gil		
TUTOR	Eduardo Juárez	VºBº	
DEPARTAMENTO	SEC		
Miembros del Tribunal Calificador			
PRESIDENTE	Javier Malagón		
VOCAL	Eduardo Juárez		
VOCAL SECRETARIO	Mariano Ruiz		
DIRECTOR			
Fecha de lectura	30/09/2014		
Calificación	El Secretario,		

RESUMEN DEL PROYECTO:

En la actualidad, nos encontramos con un gran número de sistemas empotrados con los que convivimos a diario. Estos sistemas pueden estar tanto en móviles y tecnología, como en componentes domésticos.

El Grupo de investigación - GDEM - tiene como una de sus líneas de investigación el estudio del consumo de los sistemas empotrados.

En este proyecto, elaboraremos un banco de pruebas que nos ayudará a simular dicho consumo. Para esto, crearemos una señal mediante Labview que, tras pasar por una serie de procesamientos, lograremos transferirla al sistema empotrado – Beagle Board – para su estudio.

Esta señal simulará uno de los canales de los cuales se puede componer un sistema empotrado, como por ejemplo, ejemplo si el sistema está dotado de una conexión Wifi o si utiliza una tarjeta SD externa como sistema operativo, como es nuestro caso.

RESUMEN

El Grupo de Diseño Electrónico y Microelectrónico de la Universidad Politécnica de Madrid -GDEM- se dedica, entre otras cosas, al estudio y mejora del consumo en sistemas empotrados. Es en este lugar y sobre este tema donde el proyecto a exponer ha tomado forma y desarrollo.

Según un artículo de la revista online *Revista de Electrónica Embebida*, un sistema empotrado o embebido es aquel “sistema controlado por un microprocesador y que gracias a la programación que incorpora o que se le debe incorporar, realiza una función específica para la que ha sido diseñado, integrando en su interior la mayoría de los elementos necesarios para realizar dicho función”.

El porqué de estudiar sobre este tema responde a que, cada vez, hay mayor presencia de sistemas empotrados en nuestra vida cotidiana. Esto es debido a que se está tendiendo a dotar de “inteligencia” a todo lo que puedan hacer nuestra vida un poco más fácil.

Nos podemos encontrar dichos sistemas en fábricas, oficinas de atención a los ciudadanos, sistemas de seguridad de hogar, relojes, móviles, lavadoras, hornos, aspiradores y un largo etcétera en cualquier aparato que nos podamos imaginar.

A pesar de sus grandes ventajas, aún hay grandes inconvenientes. El mayor problema que supone a día de hoy es la autonomía del mismo sistema, ya que hablamos de aparatos que muchas veces están alimentados por baterías -para ayudar a su portabilidad-. Por esto, se está intentando dotar a dichos sistemas de una capacidad de ahorro de energía y toma de decisiones que podrían ayudar a duplicar la autonomía de dicha batería.

Un ejemplo claro son los Smartphones de hoy en día, unos aparatos casi indispensables que pueden tener una autonomía de un día. Esto es poco práctico para el usuario en caso de viajes, trabajo u otras situaciones en las que se le dé mucho uso y no pueda tener acceso a una red eléctrica. Es por esto que surge la necesidad de investigar, sin necesidad de mejorar el hardware del sistema, una manera de mejorar esta situación.

Este proyecto trabajará en esa línea creando un sistema automático de medida el cual generará las corrientes que servirán como entrada para verificar el sistema de adquisición que junto con la tarjeta Beagle Board permitirá la toma de decisiones en relación con el consumo de energía.

Para realizar este sistema, nos ayudaremos de diferentes herramientas que podremos encontrar en el laboratorio del GDEM, como la fuente de alimentación Agilent y la Beagle Board -como principales herramientas de trabajo- .

El objetivo principal será la simulación de unas señales que, después de pasar un proceso de conversión y tratado, harán la función de representación del consumo de cada una de las partes que pueden formar un sistema empotrado genérico. Por lo tanto, podemos decir que el sistema hará la funcionalidad de un banco de pruebas que ayudará a simular dicho consumo para que el microprocesador del sistema pueda llegar a tomar alguna decisión.

ABSTRACT

The Electronic and Microelectronic Design Group of Universidad Politécnica de Madrid - GDEM- is in charge, between other issues, of improving the embedded system's consumption. It is in this place and about this subject where the exposed project has taken shape and development.

According to an article from the online magazine *Revista de Electronica Embebida*, an embedded system is "the one controlled by a microprocessor and, thanks to the programming that it includes, it carries out a specific function what it has been designed for, being integrated in it the most necessary elements for realizing the already said function".

Because of studying this subject, answers that each time there is more presence of the embedded system in our daily life. This is due to the tendency of providing "intelligence" to all what can make our lives easier.

We can find this kind of systems in factories, offices, security systems, watchers, mobile phones, washing machines, ovens, hoovers and, definitely, in all kind of machines what we can think of.

Despite its large advantages, there are still some inconveniences. Nowadays, the most important problem is the autonomy of the system itself when machines that have to be supplied by batteries –making easier the portability-. Therefore, this project is going after a save capacity of energy for the system as well as being able to take decisions in order to duplicate batteries' autonomy.

Smartphones are a clear example. They are a very successful product but the autonomy is just one day. This is not practical for users, at all, if they have to travel, to work or to do any activity that involves a huge use of the phone without a socket nearby. That is why the need of investigating a way to improve this situation.

This project is working on this line, creating an automatic system that will generate the currents for verifying the acquisition system that, with the beagle board, will help taking decisions regarding the energy's consumption.

To carry out this system, we need different tools that we can find in the laboratory of the group previously mentioned, like power supply Agilent and the Beagle Board – as main working tools –.

The main goal is the simulation of some signals that, after a conversion process, will represent the consumption of each of the parts in the embedded generic system. Therefore, the system will be a testing ground that simulate the consumption, once sent to the processor, to be processed and so the microprocessor system might take some decision.

ÍNDICE DE FIGURAS

Figura 1 – Esquema del Proyecto	Pag 12
Figura 2 – Frontal Fuente Agilent	Pag 15
Figura 3 – Trasera Fuente Agilent	Pag 15
Figura 4 – Explicación Frontal Agilent	Pag 16
Figura 5 – Explicación Trasera Agilent	Pag 17
Figura 6 – Configuración software Agilent	Pag 19
Figura 7 – Software Agilent	Pag 19
Figura 8 – Señal escalera Labview	Pag 22
Figura 9 – Seno 10Hz	Pag 24
Figura 10 – Seno 20Hz	Pag 24
Figura 11 – Seno 30Hz	Pag 25
Figura 12 – Driver Agilent Labview	Pag 26
Figura 13 – Función “Inicializar”	Pag 27
Figura 14 – Función “Limitar Corriente”	Pag 27
Figura 15 – Función “Output Enable”	Pag 27
Figura 16 – Función “Seleccionar tensión”	Pag 28
Figura 17 – Función “Cerrar Driver”	Pag 28
Figura 18 – Función “Creación Señal”	Pag 30
Figura 19 – Onda cuadrada 10Hz – 1 segundo	Pag 33
Figura 20 – Onda cuadrada 20Hz	Pag 34
Figura 21 – Función “Creación Señal-Configurarla”	Pag 35
Figura 22 – Seno 20Hz	Pag 36
Figura 23 – Fichero de muestras Seno 15Hz	Pag 36
Figura 24 – Primera muestra señal de 15Hz	Pag 37
Figura 25 – Segunda muestra señal de 15Hz	Pag 37
Figura 26 – Tercera muestra señal de 15Hz	Pag 38
Figura 27 – Cuarta muestra señal de 15Hz	Pag 38
Figura 28 – Quinta muestra señal de 15Hz	Pag 39
Figura 29 – Seno 30Hz	Pag 40
Figura 30 – Entrada ADC Board PSpice	Pag 44
Figura 31 – Conector ADC Board PSpice	Pag 45
Figura 32 – Circuito Howland	Pag 46
Figura 33 – Esquema KiCad	Pag 48
Figura 34 – Selección de huella KiCad	Pag 49
Figura 35 – Rutado convertidor	Pag 50
Figura 36 – Convertidor – Parte superior	Pag 51
Figura 37 – Convertidor – Parte inferior	Pag 51
Figura 38 – Entrada ADC Board – Ejemplo 1	Pag 53
Figura 39 – ADC Board	Pag 54
Figura 40 – Beagle Board	Pag 54
Figura 41 – Conexión ADC+Beagle	Pag 55

Figura 42 – Esquema Beagle	Pag 57
Figura 43 – Primera prueba de consumo	Pag 60
Figura 44 – Configuración Putty	Pag 61
Figura 45 – Configuración Putty	Pag 62
Figura 46 – Inicio Beagle	Pag 63
Figura 47 – Prueba de consumo 1	Pag 65
Figura 48 – Prueba de consumo 2	Pag 67
Figura 49 – Conexión ADC Board	Pag 69
Figura 50 – Prueba real de consumo	Pag 70
Figura 51 – Placa Prototipado 0V Entrada	Pag 71
Figura 52 – Placa Prototipado 1V Entrada	Pag 72
Figura 53 – Prueba consumo 3	Pag 73
Figura 54 – V/I – Onda cuadrada	Pag 76
Figura 55 – V/I – Onda senoidal 1	Pag 77
Figura 56 – V/I – Onda senoidal 2	Pag 77
Figura 57 – Conexión completa	Pag 78
Figura 58 – Prueba consumo onda cuadrada	Pag 79
Figura 59 – Prueba consumo onda senoidal	Pag 80

Índice

Introducción	11
Preámbulo	11
Presentación del Proyecto	12
Parte 1 - Software	13
Parte 2 - Hardware	13
Sistema Completo	13
Parte 1 – Software	15
Fuente Agilent 66321D	15
Introducción	15
Fuente de alimentación	16
Especificaciones	18
Software Agilent	18
Uso de la fuente de alimentación Agilent	20
Software Labview	21
Introducción	21
Estudio previo	21
Software Final	26
Resultados - Parte 1	32
Conclusiones	41
Parte 2 – Hardware	43
Convertidor V/I	43
Introducción	43
Convertidor V/I	44
ADC Board	53
Introducción	53
Conexión ADC	53
Beagle Board	57
Introducción	57
Esquema	57
Entorno de trabajo	59
Programación	64
Resultados – Parte 2	68
Beagle + ADC Board	69
Beagle + ADC Board + Convertidor V/I	71
Conclusiones	74
Sistema Completo	75
Introducción a las pruebas	75
Primeras conexiones	75
Conexión Sistema completo	78
Conclusiones Sistema Completo	81
Conclusiones	81
Posibles Mejoras	81

Bibliografía	83
Anexos	85
Anexo A – Labview.....	85
Anexo A.1 – Esquemas del Proyecto completo	85
Anexo A.2 – Representación Onda Seno	90
Anexo B – Convetidor V/I	91
Simulación PSpice	91
Anexo C – ADC Board.....	92
Anexo C.1 – Esquemático ADC Board	92
Anexo C.2 – Valores Resistencias.....	94

Introducción

Preámbulo

Este proyecto está basado en la línea de investigación para la optimización de energía. Dicha investigación está desarrollada por el Grupo de Diseño Electrónico y Microelectrónico -a partir de ahora “GDEM”- de la Universidad Politécnica de Madrid. Con él se pretende desarrollar un sistema capaz de simular las corrientes de consumo de un sistema empotrado mediante el software Labview.

Dicha línea de investigación conlleva dos partes claramente diferenciadas: 1. La simulación comentada anteriormente; 2. El estudio del consumo en el procesador, para poder ver cómo podemos pasarle la información y dotarle de unos procesos capaces de tomar decisiones para optimizar su periodo de actividad.

Dada la extensión de toda la investigación en su conjunto, este proyecto a presentar se centrará únicamente en la primera parte, dejando así vía libre para otros proyectos que la completen.

La idea es la realización de un banco de pruebas en el cual podamos generar una señal, mediante una fuente de alimentación, que nos servirá para que el procesador del sistema empotrado pueda leerla y, en función de los resultados, poder realizar las modificaciones necesarias para poder mejorar su autonomía minimizando el consumo.

Como primera parte, nos encargamos de la generación de la señal que se enviará al sistema. Esto lo podremos realizar gracias a una fuente de alimentación programable mediante labview, donde elaboramos un programa capaz de generar una señal escrita anteriormente ya sea en un fichero o generándola automáticamente por Labview. Dicha señal, será convertida a corriente gracias a un convertidor - que hemos realizado en este proyecto – para que el procesador del sistema empotrado pueda procesarla. Para llevar a cabo este proceso, nos ayudamos de un proyecto [12] realizado por el GDEM en el año 2012 - *Peripheral SPI use with OMAP3530 – Advanced digital Architectures* por Miguel Chavarrias – En el cual se elabora una placa capaz de medir señales recibidas mediante un amplificador de instrumentación y pasarlas a un convertidor ADC conectado a los canales SPI de una Beagle Board.

Para poder llevar a cabo el proyecto, haremos una breve explicación de los componentes que vamos a utilizar:

- PC: Instalado con el software de National Instrument nos ayudará a simular las señales deseadas.
- Fuente de Alimentación: El modelo Agilent 66321D es una fuente programable con Labview que nos servirá para sacar la señal programada en dicho software.
- Convertidor V/I: Una placa analógica que usaremos para convertir la tensión recibida de la fuente a corriente.

- ADC Board: del proyecto “*Advanced digital Architectures*” –al cual nos referiremos como “Proyecto ADA”, donde podremos pasar dicha señal al procesador mediante un amplificador de instrumentación y un ADC.
- Beagle Board: donde pasaremos la información al procesador y nos la mostrará por pantalla.

Una vez descritos los componentes, su esquema de conexión será el siguiente:

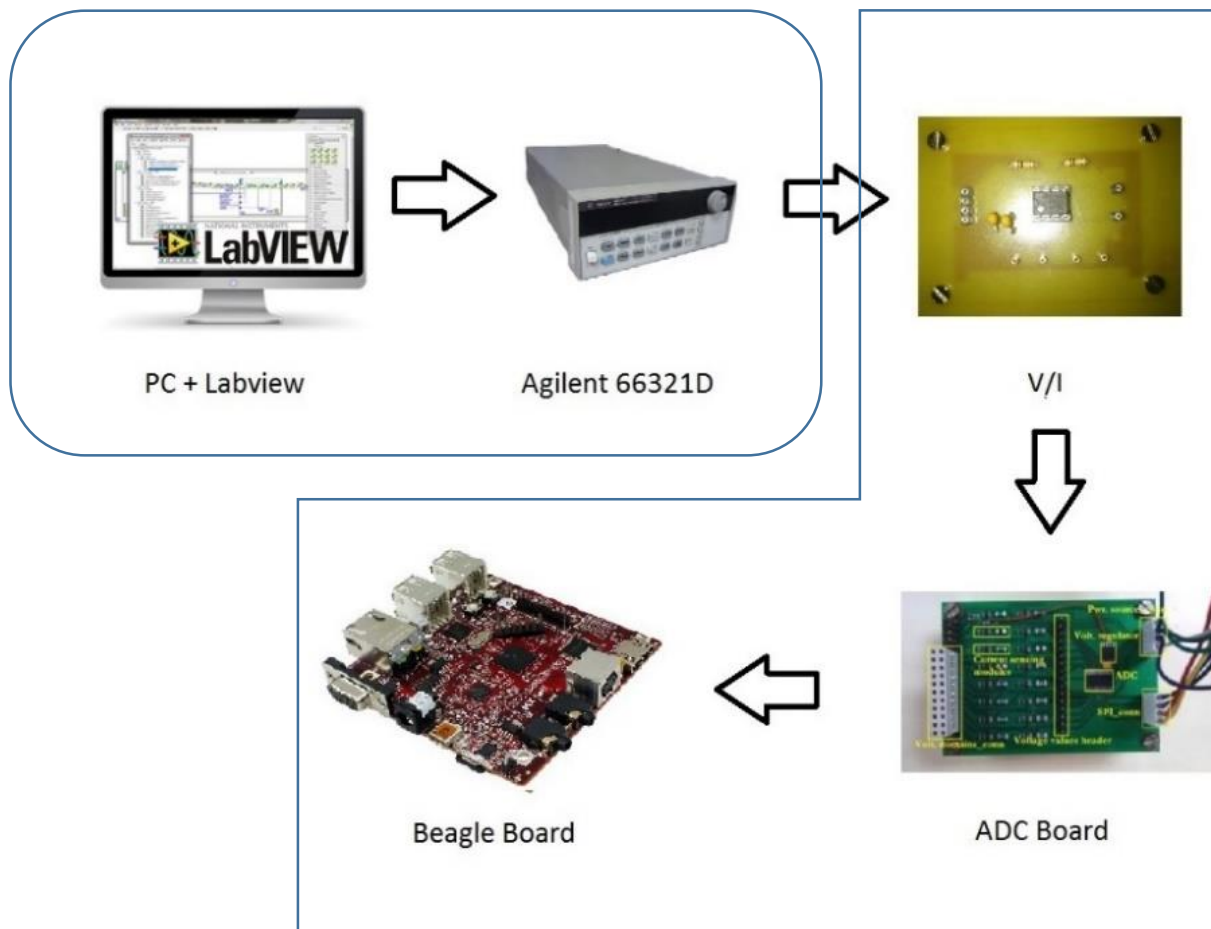


Figura 1

Presentación del Proyecto

Vamos a explicar cómo está desarrollado este documento para una mayor comprensión de su estructura.

Nos ayudaremos del esquema que hemos puesto en el preámbulo -Figura 1-, donde vemos que el proyecto se puede dividir en dos partes:

Parte 1 - Software

Esta parte se compone de la Fuente de Alimentación Agilent y el Software de Labview. En ella, explicaremos lo necesario para generar una señal de cualquier tipo y poder sacarla por la fuente de alimentación seleccionada.

En teoría, la generación de esta señal, nos ayudará a simular el consumo de cada uno de los canales de un sistema empotrado.

También, podremos ver los puntos fuertes de la fuente y por qué hemos elegido el Software de Labview cuando el propio fabricante tiene un software propio para el manejo de dicha fuente.

Al final de este apartado nos encontraremos con una conclusión y unas capturas mostrando los resultados que hemos podido obtener.

Parte 2 - Hardware

Esta parte es algo más extensa que la anterior. En ella, podremos ver cómo tratamos la señal de la parte 1 para finalmente acabar en la Beagle Board.

Lo que vamos a utilizar para el desarrollo de esta parte es una placa creada por nosotros mismos con un software libre -llamado KiCad- que nos servirá de convertidor, una placa del Proyecto ADA – ADC Board - y una Beagle Board.

Con esto, tenemos que enviar la señal desde la fuente de alimentación al ADC. Esta conversión analógico-digital podrá pasar a la Beagle Board que, gracias a un programa en C que tomaremos del “Proyecto ADA”, podremos ver la señal recibida en pantalla.

Una vez conseguidas las premisas de este apartado, ya seremos capaces de pasar información de la señal simulada en la Parte 1 al procesador de la Beagle Board.

También aquí, habrá unas conclusiones con sus pruebas y resultados correspondientes para verificar las partes por separado.

Sistema Completo

Una vez comprobadas las dos partes de las que se compone el proyecto, pasaremos al apartado final, donde veremos el sistema completo, cómo están conectadas las diferentes partes y las pruebas necesarias para verificar que el sistema funciona como esperamos.

También nos encontraremos con un apartado donde veremos sus limitaciones y sus posibles mejoras, ayudándonos de unas capturas por mejor visualización del mismo.

Una vez finalizado, tendremos nuestro banco de pruebas completo, pudiendo realizar la simulación de un canal y ver la información en la Beagle.

Parte 1 – Software

Fuente Agilent 66321D

Introducción

El modelo Agilent 66321D es una fuente de alimentación de alto rendimiento que nos proporciona medidas con gran rapidez y una excelente respuesta transitoria de tensión. En dicho modelo, también podemos disponer de una entrada auxiliar de medida de tensión DVM, que se corresponde a un voltímetro digital, que además es programable desde el panel frontal usando los comandos de programación de esta fuente (SCPI). También nos encontramos con una interfaz de conexión GPIB IEEE-488, con la que podremos conectar dicha fuente al PC. [1]

Características generales:

- Tensión, corriente y resistencia interna programables, con una resolución de 12bits
- Gran cantidad de modos de medida:
 - Medida DC.
 - RMS y valores pico.
 - 16 bits de resolución
 - Posibilidad de captura mediante disparo.

Para hacernos una idea de las características de la fuente, vemos un par de imágenes donde mostramos dicha fuente y el cable GPIB que usaremos para conectarnos al ordenador.



Figura 2



Figura 3

Fuente de alimentación.

- Panel Frontal [1]

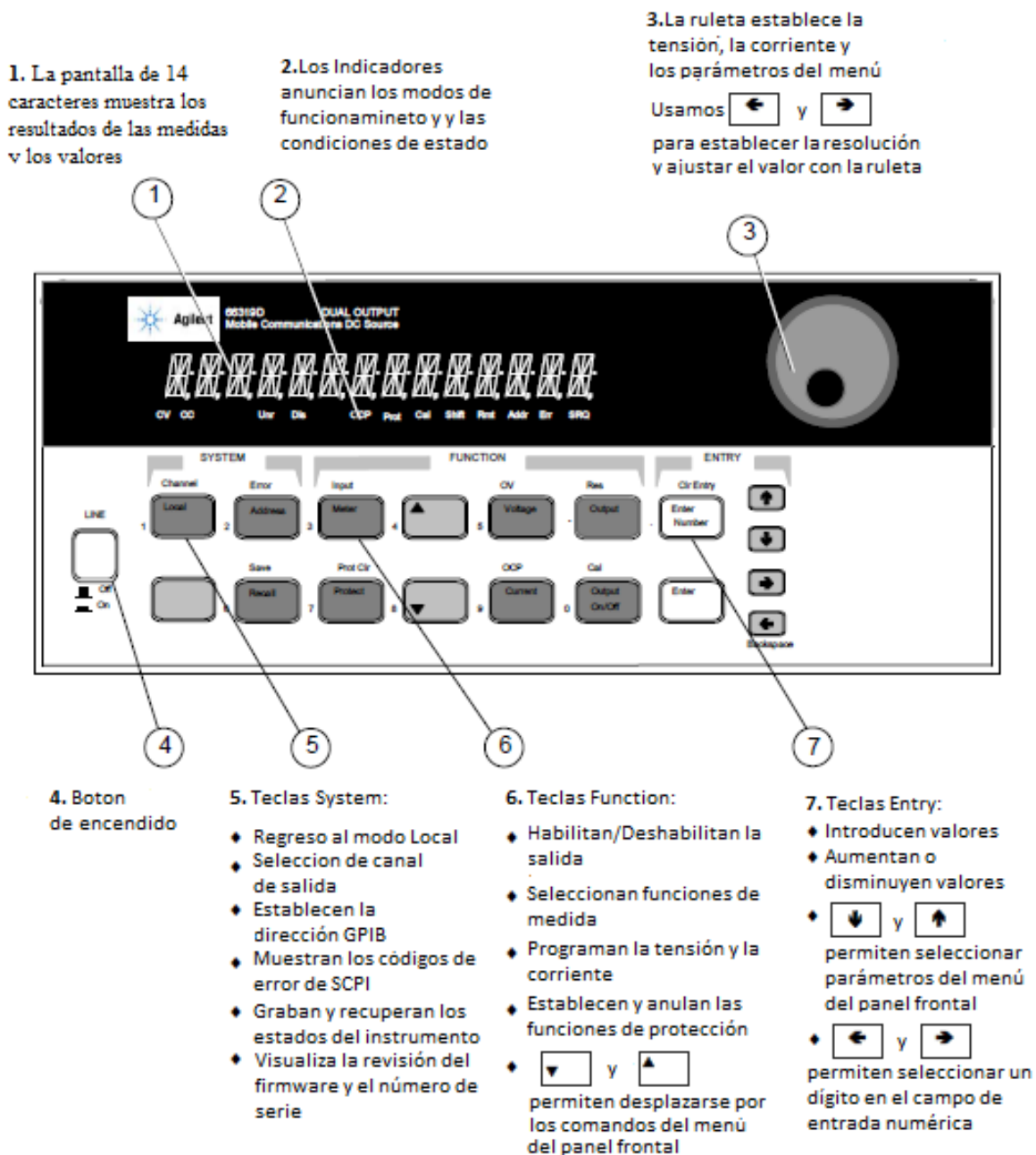


Figura 4

- Panel Trasero. [1]

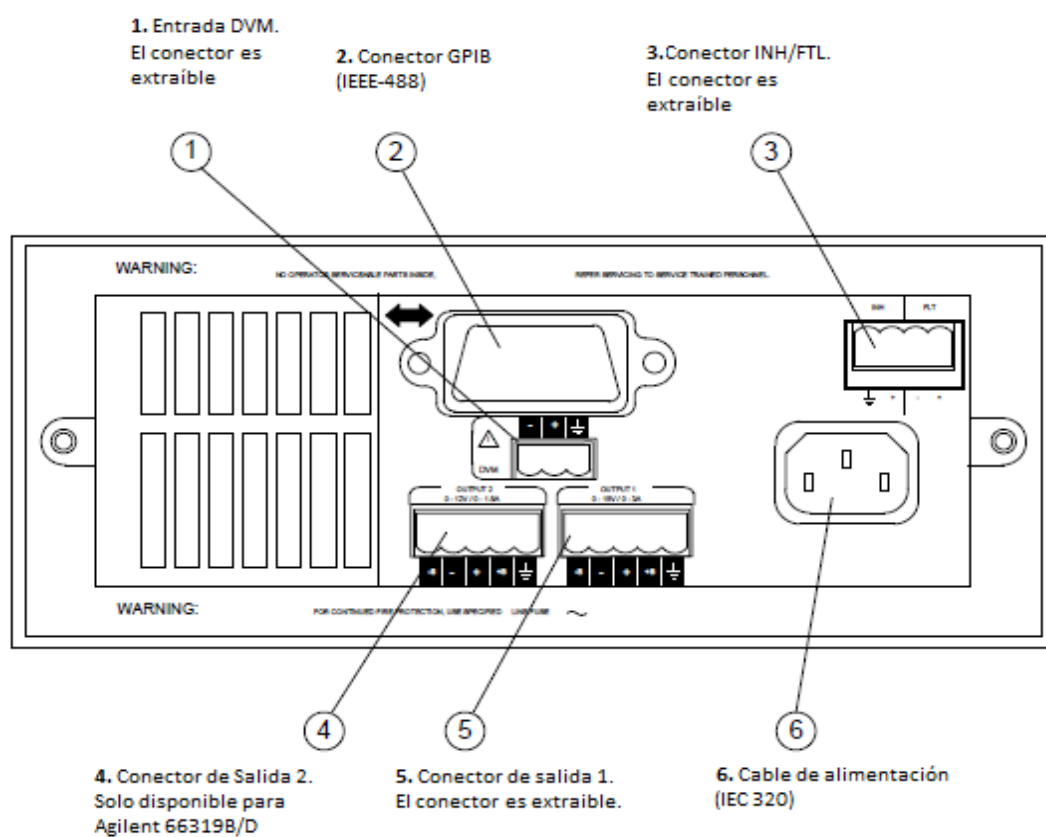


Figura 5

Especificaciones

Para poder llevar a cabo nuestro proyecto, tenemos que tener en cuenta las especificaciones que debemos cumplir para un correcto funcionamiento del equipo.

Entonces, usaremos la fuente de alimentación para simular una señal. Lo más importante, es tener en cuenta los valores máximos y mínimos de tensión y los tiempos de refresco de dicha señal. Los valores más importantes son: [1]

- Un valor de tensión en la salida de 0 a 15V.
- Un valor de corriente en la salida de 0 a 3A
- Una precisión en la programación de 0.05% + 10V y 0.05% + 1.33mA.
- Un ruido de 1mV/6mV(rms/p-p)
- Un valor de refresco máximo de 10ms en la salida.

La principal característica de la fuente de alimentación, es que es programable por ordenador mediante la interfaz GPIB. En nuestro caso nos ayudaremos, para realizar dicha programación, de un software de National Instrument llamado Labview.

Respecto al uso de la fuente, no necesitamos una gran precisión de las señales que vamos a sacar, puesto que como veremos, las señales tendrán una frecuencia baja y podremos ver cómo es más fácil que se establezca el valor que tenemos que sacar por la fuente. Por lo que con las características apuntadas de la fuente son suficientes para desarrollar nuestro trabajo.

En el siguiente apartado, explicaremos cómo llevaremos a cabo esta programación y mediante qué driver es posible

Software Agilent

La empresa Agilent ofrece la posibilidad del uso de un software para el manejo de la fuente de alimentación. Éste software [2] tiene unas funciones básicas para manejar el entorno, como por ejemplo, la medida de señales, la programación de señales continuas en la salida, un visor tipo osciloscopio para el uso del voltímetro.

En nuestro caso, nos interesa la generación de señales de varios tipos: cuadradas, sinusoidales, triangulares...

También es interesante la importación de señales de cualquier tipo mediante ficheros de texto, que nos ayudaran a hacer de nuestro sistema un poco más flexible a la hora de interactuar con otros entornos de programación.

Por esto, nos vemos obligados a utilizar una herramienta de National Instrument llamada Labview, con la cual tenemos una mayor libertad de programación y podemos dejar abierto su desarrollo para posibles mejoras que se puedan ir incorporando.

Hemos instalado el programa facilitado por el fabricante para poder realizar unas primeras pruebas con la fuente de alimentación.

Para proceder la instalación, seguimos una serie de pasos que encontramos en el manual de usuario que nos facilita el fabricante [2]:

- Inserción del CD que viene en la caja de la fuente.
- Selección de los directorios de instalación
- Ejecutar la aplicación Agilent 14565B

Una vez arrancado el programa lo primero que nos pide es su configuración, como vemos en la Figura 6:

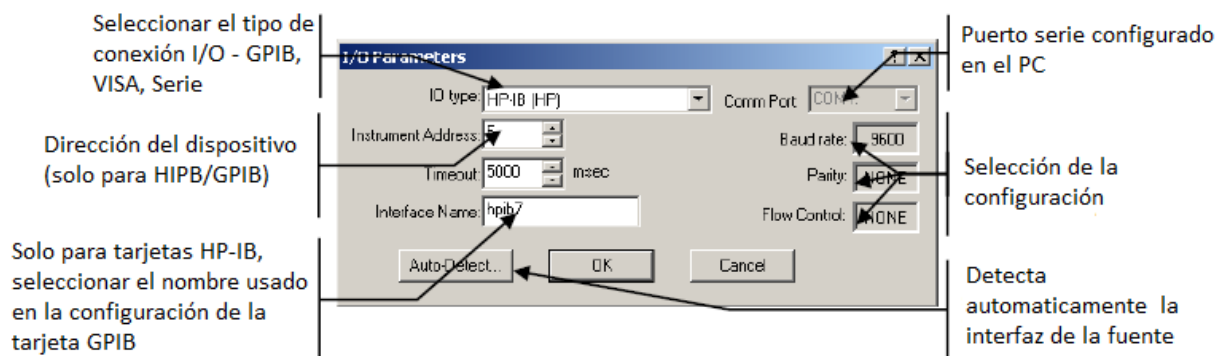


Figura 6

Una vez instalado y configurado el software, tendremos una ventana como la Figura 7, donde podremos observar las posibilidades que tenemos con dicho software y que explicaremos a continuación.

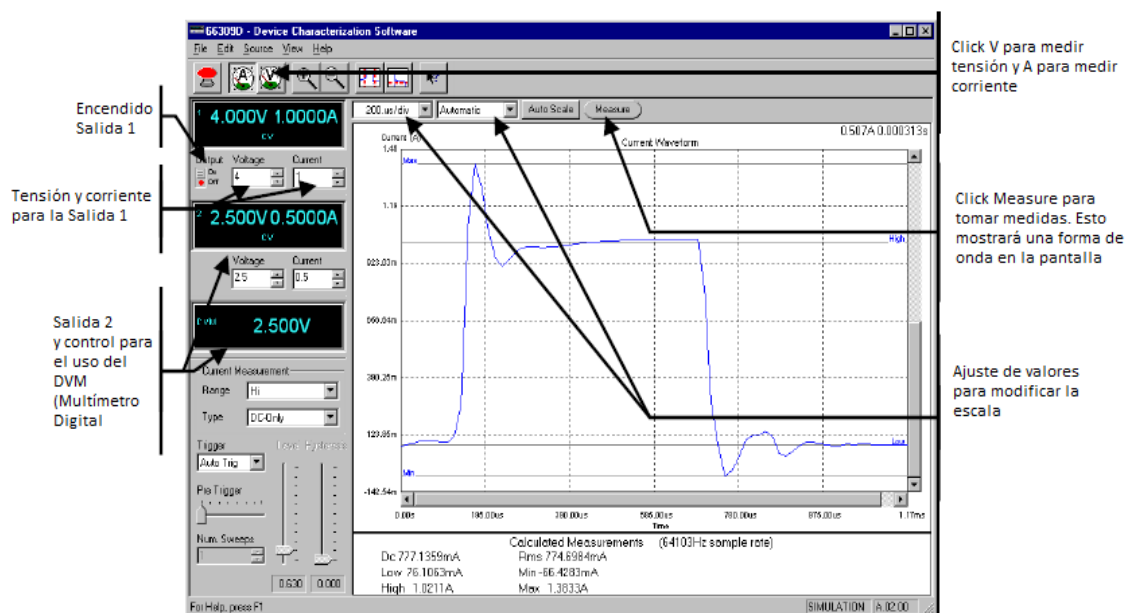


Figura 7

En esta captura, vemos una interfaz sencilla y simple, donde podemos:

- Fijar los valores de tensión deseados.
- Usar un osciloscopio virtual que muestre los valores obtenidos, dándonos varias opciones de medida bastante completas.
- Seleccionar la salida deseada -aunque nuestra fuente tan solo este dotada de una-.
- Usar el Multímetro digital del que está equipado en la fuente.

Como ya sabemos, el software elegido en nuestro proyecto es el Labview. Tomamos esta decisión por la versatilidad que nos aporta este software. Agilent no nos permitiría la importación de señales desde un fichero, ni nos serviría para la conexión de otras fuentes de alimentación. Ya que estamos realizando un banco de pruebas, queremos dotar a éste de un gran abanico de posibilidades, en las que se pueda mejorar nuestra parte del proyecto sin ser necesario el uso de este modelo de fuente.

Sin embargo, es importante instalar dicho software para realizar unas primeras pruebas y hacer una toma de contacto con la fuente. Hemos de saber cuál es el canal de comunicación así como son sus principales características.

[Uso de la fuente de alimentación Agilent](#)

Ahora que conocemos las partes de las que se compone nuestra fuente de alimentación, conozcamos el uso que le vamos a dar, puesto que, más adelante con el programa realizado en Labview, podremos comprender el entorno completo.

Anteriormente, hemos comentado que la fuente dispone de una salida de tensión, de un voltímetro digital y de una salida GPIB.

En nuestro proyecto, usaremos dicha salida de tensión para sacar de ella una señal que programaremos en el entorno de Labview y la mandaremos a través de la interfaz GPIB.

En nuestro caso, al disponer del modelo 66321D, sólo podremos mandar la información de una señal y, por lo tanto, de un solo canal para simular el consumo del sistema. En futuros proyectos, se podría investigar la forma de sacar 16 canales independientes de dicha fuente de alimentación y así simular el sistema empotrado completo tal y como viene definida en el proyecto ADA.

Software Labview

Introducción

Como ya adelantamos en la introducción de la fuente de alimentación, Agilent nos da un software que no es suficiente para la realización de nuestro proyecto y, por ello, nos vemos obligados a realizarlo mediante Labview.

Labview [3][4][5] es una plataforma y entorno de desarrollo para diseñar sistemas con un lenguaje de programación visual gráfico. Es muy apropiado para sistemas hardware y software de pruebas, control y diseño, simulado o real y embebido.

Los programas desarrollados con Labview se llaman Instrumentos Virtuales o Vis- como nos referiremos a partir de ahora-, y su origen provenía del control de instrumentos, aunque hoy en día se ha expandido ampliamente no sólo al control de todo tipo de electrónica - Instrumentación electrónica- sino también a su programación embebida, comunicaciones, matemáticas, etc.

Elegimos Labview por la gran variedad de integración Hardware que nos ofrece. En nuestro caso, podemos descargar, de la misma web del fabricante, todos los driver necesarios para un completo manejo de cualquier fuente Agilent.

Otra de las características principales que nos ofrece y utilizaremos en nuestro proyecto es la interacción entre estos. La característica que antes habíamos denominado como VI, nos permite crear un entorno para nuestro programa formado por varios programas, o VIs, que nos ayudará a crear un software dividido en varios Vis que nos llevará a una mayor eficacia a la hora de la resolución de problemas y dotará a nuestro sistema de una gran flexibilidad, pudiendo añadir varios Vis, e incluso, usar Vis de otras personas que puedan desarrollar nuestra investigación.

Estudio previo

Nuestro proyecto debe de ser capaz de sacar por la fuente de alimentación cualquier señal de entrada que necesitemos. Para ello, vamos a estudiar cómo podremos muestrear la señal a representar y cuáles son las limitaciones que nos encontraremos al usar dicha señal.

En primer lugar, tendremos que saber cuál es el tiempo mínimo de variación de señal para nuestra fuente de alimentación, teniendo en cuenta que es una fuente en régimen continuo y no está pensada como generador de ondas.

Para ello, sacamos una señal que varíe en el tiempo lo más deprisa que podamos:

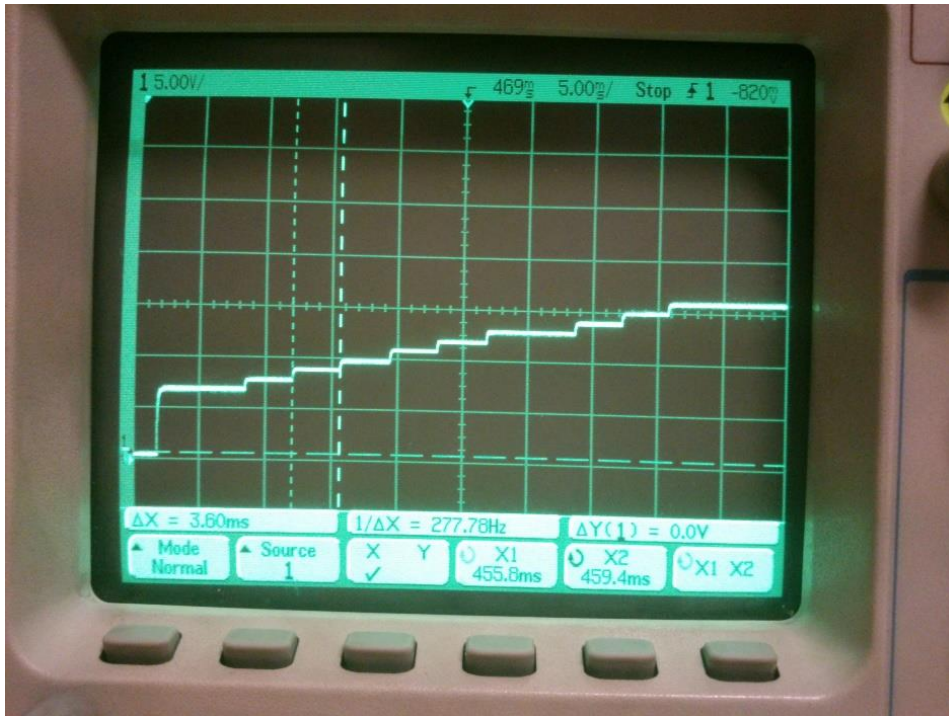


Figura 8

Aquí podemos observar un cambio máximo de unos 8ms y un mínimo de unos 5ms.

Tomando estas referencias, cogemos el más lento para no correr peligro de deformar la señal y decidimos generar una señal que saque una muestra como mínimo de 10ms

Para hacernos una idea, en la introducción adelantábamos que íbamos a estudiar 2 tipos de señales características: una onda cuadrada y una onda senoidal. Hemos realizado un estudio teórico de cuáles serán nuestras limitaciones a la hora de representar una onda de este tipo. Siempre teniendo en cuenta las limitaciones de la fuente.

Por lo tanto, como señales mínimas y máximas nos quedarían las ondas que vamos a representar en las Figuras 9, 10 y 11, ayudándonos de una hoja de cálculo donde muestreamos las señales deseadas y las mostramos en un gráfico.

Para la preparación de esta hoja de cálculo, hemos utilizado las funciones necesarias para sacar los valores importantes y hemos representado las gráficas que adjuntaremos en el documento. Para ver la hoja de cálculo, podemos ir al Anexo A.2, donde veremos las señales muestreadas.

Las ondas quedarían:

- Onda cuadrada: 50 Hz

Es la onda con la frecuencia máxima que podemos sacar. Corresponde al valor que habíamos fijado de muestro, es decir, 10 ms para una muestra y 10 ms para la siguiente. Consiguiendo un periodo de 20 ms y una frecuencia de 50 Hz.

Si queremos sacar una señal con una frecuencia mayor, la fuente no es capaz de superar dicha velocidad, por lo tanto saltaremos muestras y falsearíamos la señal.

Más tarde, en el apartado de “Conclusiones”, analizaremos y realizaremos algunas de estas pruebas y veremos cuál sería la respuesta real de nuestro sistema ante estos factores.

Para la onda seno, nos valdremos de un VI distinto al de la onda cuadrada, por lo que tendrá sus propias limitaciones como veremos en el siguiente estudio teórico.

- Onda senoidal: 10Hz, 20Hz y 30Hz.

Con la misma limitación, solamente podremos sacar una muestra cada 10 ms, por lo tanto, solo podrá verse una señal que tenga el doble de frecuencia, por el teorema de Nyquist. Si tomamos una onda de 30 Hz, tendremos una muestra a los 10ms, 20ms y 30ms, lo que hace que no podamos averiguar de qué onda se trata.

Las Figuras 9, 10 y 11, muestran cómo quedaría un estudio teórico de dichas frecuencias. La onda que veremos en la fuente de alimentación será una onda digitalizada donde, a menor frecuencia, apreciaremos menos dicho efecto.

Empezaremos con la onda de 10 Hz. Al muestrear dicha onda a 100 Hz, tendremos un valor cada 10ms, por lo tanto tendremos 10 muestras por cada periodo de la onda. Unas muestras más que suficientes para poder estudiar dicha señal.

Recordamos que éste es un estudio teórico. Las señales reales, se mostrarán en el siguiente apartado de “Conclusiones”.

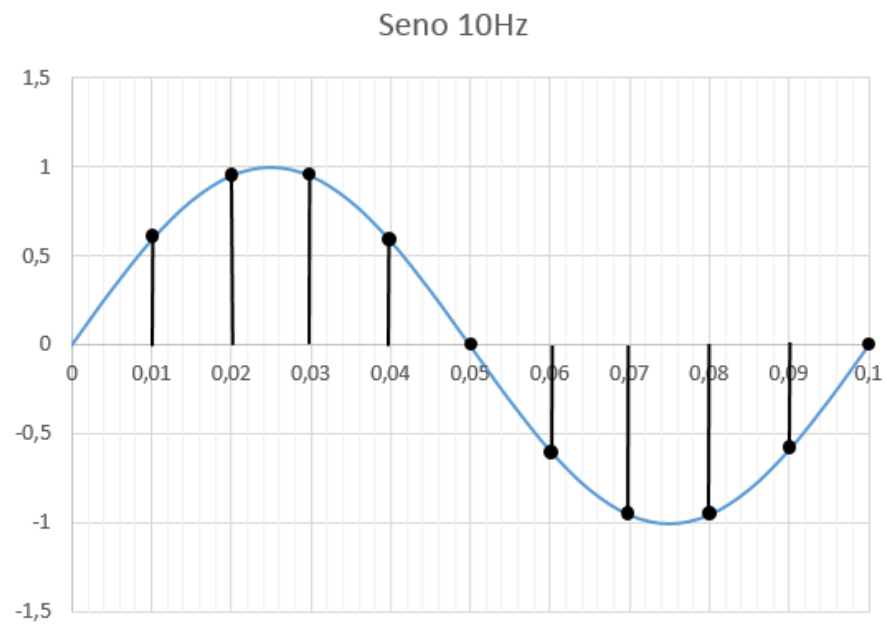


Figura 9

Seguimos con una onda de 20Hz, muestreada a 100Hz, como habíamos comentado antes.

En este caso, empezamos a tener menos valores, pero al tener 5 muestras por periodo es posible construir la señal como una onda periódica y senoidal. Ya veremos cómo se va degradando la señal a partir de dicha frecuencia, fijando esta señal como onda máxima para una mayor seguridad de representación de la señal.

Como veíamos en la anterior señal, con 5 muestras por periodo, es posible representar la señal de una manera clara. Se apreciará más el problema de la señal digital, pero todavía se sigue viendo una señal periódica y senoidal.

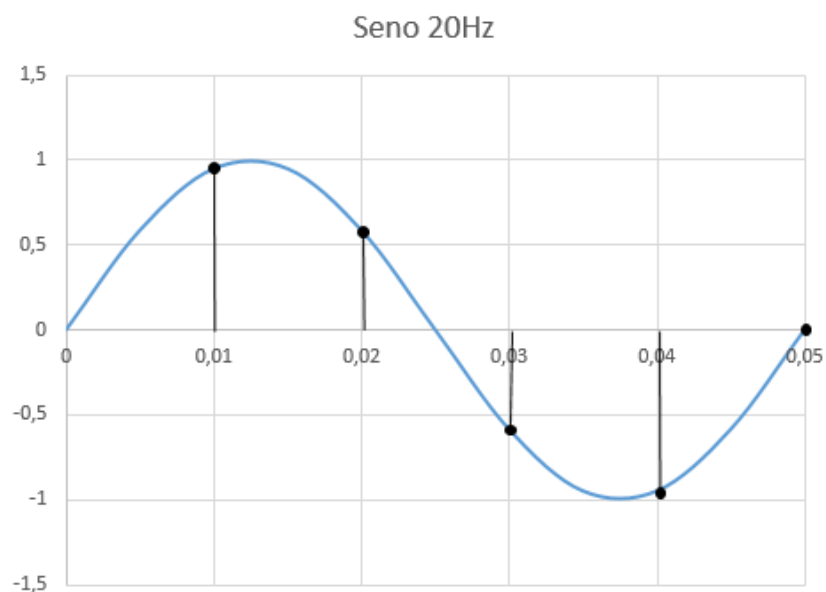


Figura 10

Por último, estudiamos la señal de los 30Hz, lo que ha sobrepasado la barrera de Nyquist. Podemos ver, que con 3 muestras por periodo, sería imposible reconstruir la señal.

Gracias a la hoja de cálculo, se aprecia de una forma sencilla esta representación y vemos ya en el muestreo que dicha señal no se correspondería con una señal senoidal. Señalamos en la gráfica los valores a los que se corresponderían la señal de muestreo de 100 muestras/s y vemos que no sigue un orden lógico y por lo tanto queda totalmente descartada.

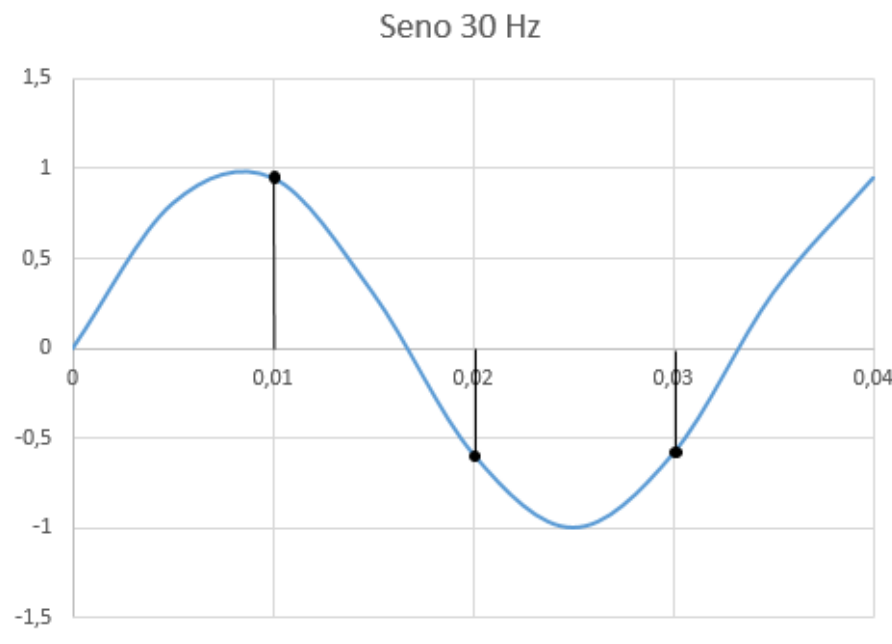


Figura 11

Software Final

Driver

Para poder trabajar con la fuente Agilent en Labview, necesitamos el driver que el mismo fabricante nos facilita. Una vez instalado, nos quedara algo como la figura siguiente: [6]

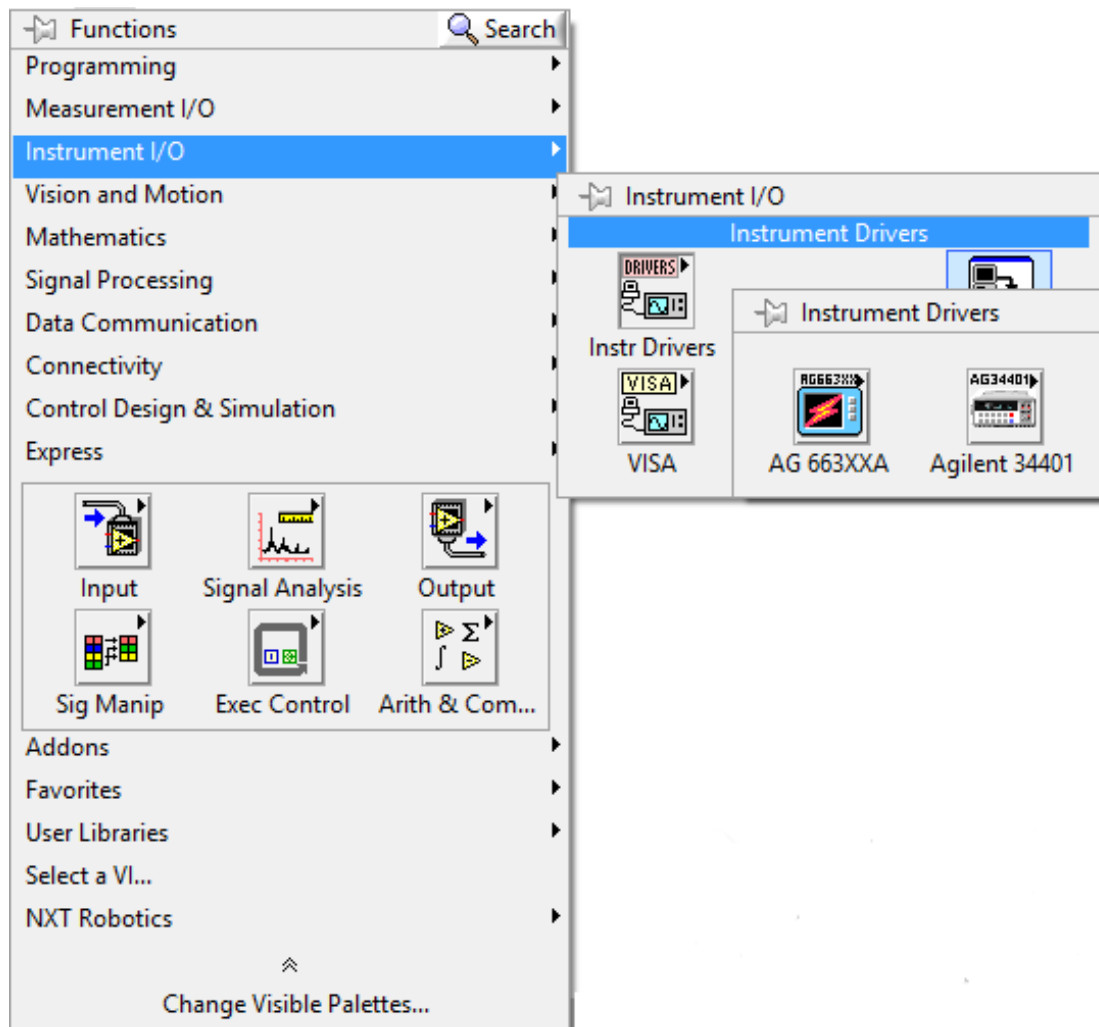


Figura 12

Para poder realizar este programa, nos hemos ayudado de las funciones que nos ofrece el driver que nos da el fabricante.

Vamos a usar las funciones más básicas para el uso de la fuente de alimentación. Tales como inicializarse, cerrarse y fijar corrientes y tensiones.

Este driver trae consigo una gran cantidad de funciones que no son útiles para nuestro proyecto, pero que cabe destacar para futuras mejoras que se pudieran realizar.

Vamos a analizar las funciones utilizadas:

- Inicializar la fuente



Figura 13

Esta función nos sirve para inicializar el driver, tan solo le tenemos que indicar dónde queremos crear la nueva sesión, que en nuestro caso, nos la marca por defecto, usando la interfaz GPIB, como vemos en la imagen.

- Limitador de corriente

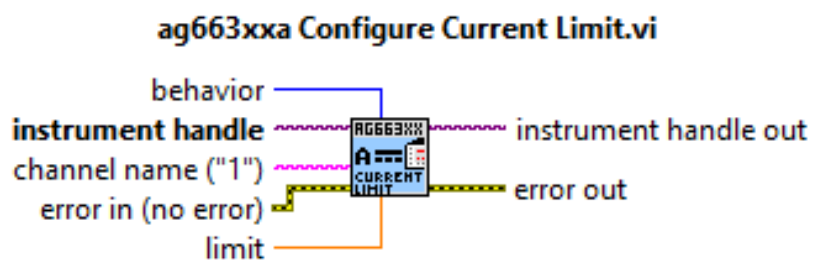


Figura 14

Hemos decidido usar el limitador de corriente, que cortará la fuente de alimentación una vez pasado el límite elegido. Para asemejarlo al comportamiento de una fuente real, hemos elegido usar un Dial para fijar dicho valor.

- Habilitar la fuente

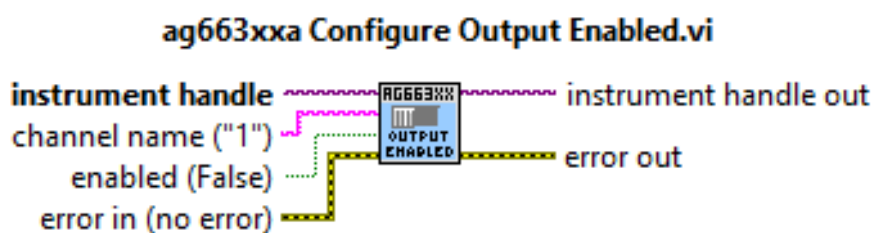


Figura 15

Al igual que en el apartado anterior, queremos dotar al programa de funciones mínimas que realiza una fuente de tensión. Utilizamos esta función para habilitar la salida de tensión en la salida.

- Entrada de tensión

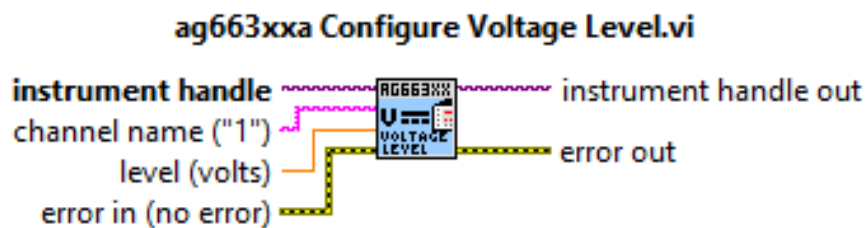


Figura 16

Con esta función, podemos sacar la tensión que queramos por la fuente de alimentación, con una posibilidad de refresco de aproximadamente 10ms.

- Finalizar sesión del driver

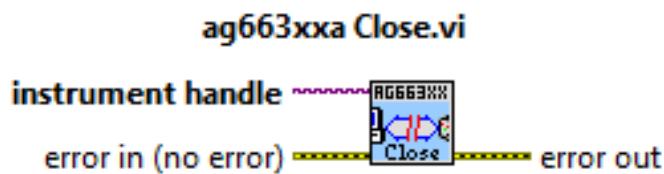


Figura 17

Para cerrar la sesión del driver, tenemos que valernos de esta función, que finalizará los procesos de la fuente de alimentación.

Programa

El programa que tenemos que realizar tendrá la opción de elegir entre dos tipos de señales: Cuadrada y senoidal.

Es por esto que tenemos una ventana principal donde debemos seleccionar qué tipo de señal queremos y cuál será su frecuencia. Como diferenciamos estas dos versiones, tenemos dos proyectos unidos en una ventana principal.

El programa está pensado para mostrar una señal de 5 segundos -suficientes para poder verificar su funcionamiento-. Pero este tiempo se puede cambiar fácilmente, ya que es una variable del mismo programa.

En nuestro programa nos encontramos con una pantalla principal pensada para elegir si queremos mostrar una señal cuadrada o senoidal, seleccionar su frecuencia, habilitar y deshabilitar la fuente.

En el “Anexo A.1 – Sistema completo” encontraremos la pantalla principal que es un entorno el cual consta de dos VIs unidos en uno: Onda cuadrada y Onda senoidal.

- Onda Cuadrada

En el VI de la onda cuadrada, escribimos en un fichero de texto los valores de amplitud que queremos que muestre la fuente de alimentación. Más tarde, leemos el fichero y mostramos los valores con un time delay igual a la mitad del periodo de la señal, quedando una señal cuadrada completa. Gracias a Labview, podemos elegir de una manera simple el tamaño del fichero y por ende la duración de la señal que queremos sacar.

El hecho de leer de un fichero la amplitud es para dotar al programa de una mayor flexibilidad a la hora de importar señales.

El programa completo lo podemos ver en el “Anexo A.1 – Sistema completo” de este documento. Como vemos en el documento, usamos una programación por etapas en Labview.

Primero inicializamos el driver de la fuente de alimentación; después, procedemos a limitar la fuente y habilitar la salida para leer del fichero seleccionado los valores de tensión; a continuación podemos fijar el tiempo de duración de la onda multiplicando el valor del bucle while por el de time delay de la función; y finalmente, cerramos el driver que se encargará de finalizar los procesos de la fuente de alimentación.

- Onda Seno

Este programa es un poco más completo que el anterior. Ya que no se trata de sacar simplemente dos muestras, si no de escribir una señal senoidal muestreada a 100 muestras/s, con sus correspondientes valores, para más tarde sacarla por dicha fuente.

La principal diferencia, es la creación de la señal. Para esto, usamos una función existente en Labview: [6]

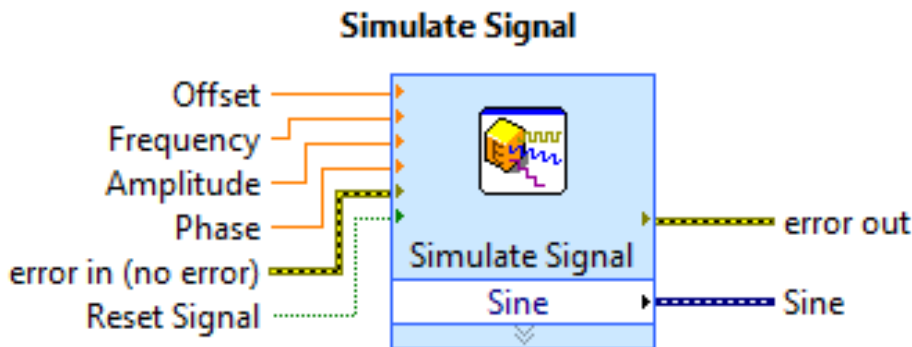


Figura 18

Como vemos, podemos crear la señal eligiendo varios de sus parámetros.

Para nuestro proyecto, nos es suficiente con elegir dos de estos: Amplitud y frecuencia. Puesto que lo que nos interesa es sacar una señal, sin importarnos el offset o la fase que tenga.

Otra de las posibilidades de esta función y que es aplicable a nuestro proyecto es la elección del tipo de Onda, siendo: cuadrada, triangular, de diente de sierra y continua.

Es importante comentar el por qué no hemos utilizado esta función para la creación de una onda cuadrada y los motivos son simples: básicamente, queremos tener la posibilidad de escribir en un fichero el tipo de señal que queremos sacar y poder variarla o importarla a nuestro antojo. Otro de los motivos es el tamaño del programa, una onda cuadrada se basa en dos muestras, ¿por qué es necesario escribir un fichero de 100 muestras si nos basta con tan solo dos?

Este VI es muy parecido al anterior, tan solo se le añade la primera etapa de creación de la señal, donde con la función que acabamos de analizar, seremos capaces de crear la señal deseada.

El siguiente paso es más parecido al anterior. Se trata de leer el fichero y escribir -a la frecuencia de muestreo que se ha escrito la onda creada- la señal para sacarla por la fuente de alimentación. Una vez más, para poder observar el programa completo, ha de dirigirse al “Anexo A.1 –Onda Seno”

Con la explicación de los dos proyectos que hemos creado, pasamos a explicar la ventana principal, donde seleccionará el usuario final los valores deseados.

Como vemos en el Anexo, nos ayudamos de una simple estructura IF para elegir dichos valores.

Hemos agregado al panel una palanca para seleccionar entre señal cuadrada y señal senoidal. Hemos habilitado un control booleano para habilitar o deshabilitar la salida de la fuente. Y por último, hemos habilitado un control numérico para seleccionar la frecuencia de la señal deseada, para los dos tipos de señales.

A continuación, en el apartado de resultados, podremos ver las señales reales de las que hablábamos anteriormente, las limitaciones a las que se enfrenta la fuente de alimentación y cómo hemos interconectado la fuente al ordenador para llevar a cabo su programación.

Resultados - Parte 1

Una vez explicadas las partes de las que se compone este primer apartado, hemos procedido a realizar las pruebas pertinentes para comprobar su funcionamiento.

Para realizar estas pruebas, el primer dato en el que nos hemos fijado es en el mencionado en el apartado de la fuente de alimentación, donde vemos que la fuente tiene una capacidad de sacar una muestra cada 8 ms como valor máximo.

Para darle un margen de seguridad, ligeramente mayor al tiempo establecido por la fuente, nos aseguramos una señal con una frecuencia de muestreo de 100 muestras/s, capaz de sacar un dato cada 10ms.

Así pues, a la hora de escribir una señal en un fichero, tenemos que recordar a la frecuencia a la que ha sido muestreada la señal. Es importante tener en cuenta este tiempo, ya que Labview, saca un los valores a la velocidad que le permite el programa y estaríamos sacando una señal que no se corresponde con la realidad.

La primera prueba realizada es la de la onda cuadrada, ya que el programa es algo menos complejo que el de la onda senoidal.

Como adelantábamos antes, la máxima frecuencia de la señal que la fuente nos permite sacar es un valor cada 10 ms, por lo tanto tendremos una frecuencia de 50 Hz.

En la siguiente prueba estuvimos más limitados, pudiendo mostrar como máximo una frecuencia de 20Hz.

Vamos a explicar cómo tomamos las ondas de cada tipo y vamos a adjuntar unas capturas para una mayor visualización del proceso.

- Onda Cuadrada:

Es el ejemplo más fácil de realizar, como adelantábamos antes, leemos de un fichero la amplitud de la señal y mediante un delay seleccionamos su periodo. Este fichero tan solo consta de dos muestras, pero podemos importar cualquier señal muestreada previamente. Recordamos que el Labview saca cada muestra leída a la velocidad que estipula el programa, por lo que nos vemos obligados a fijar un delay que fija el valor del periodo que queramos para nuestra señal, nunca inferior de 10 ms, puesto que tendríamos el problema de saltos en las muestras de la señal representada.

Vemos unos ejemplos de señales cuadradas:

1) Señal de 1s, 10Hz

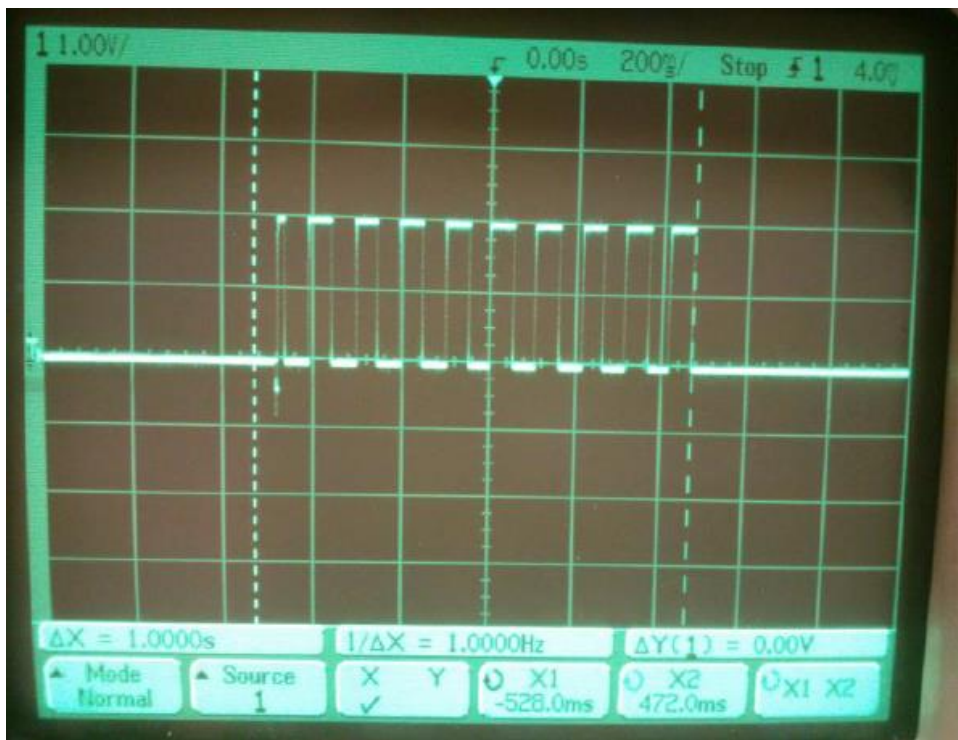


Figura 19

2) Señal de 1s, 20Hz

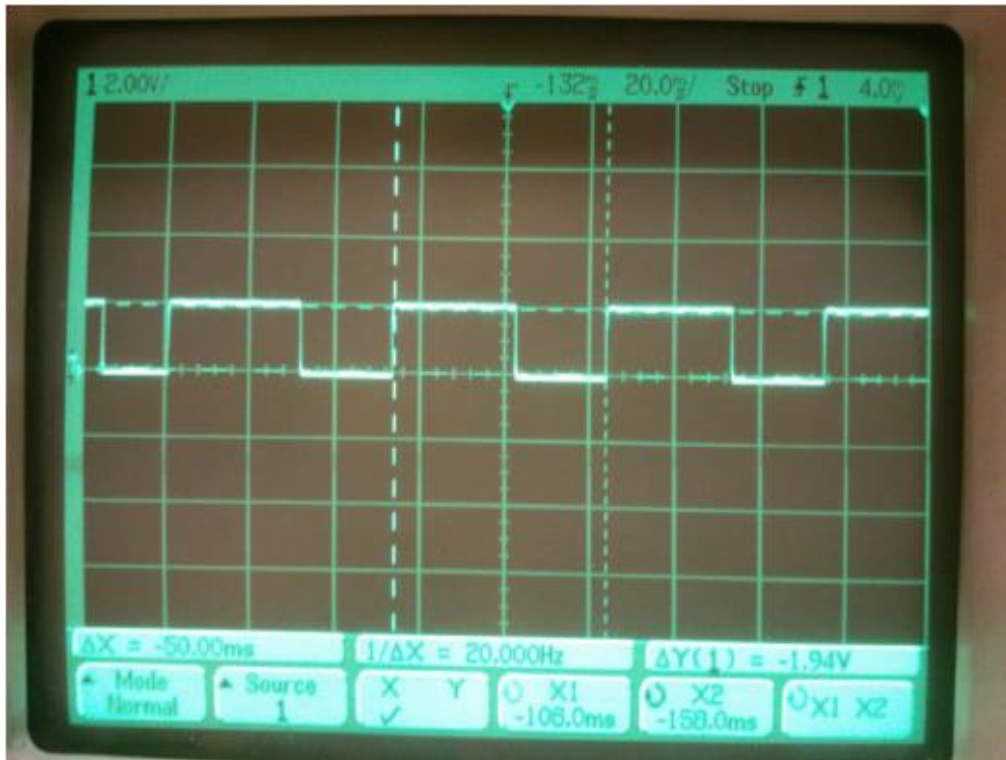


Figura 20

- Onda senoidal:

Para poder generar una onda senoidal, utilizamos una función de Labview capaz de generar cualquier señal que queramos y de cualquier tipo, como explicábamos anteriormente en las funciones utilizadas en Labview. En la Figura 21 de la página siguiente observamos la facilidad con la que se pueden crear señales, pudiendo modificar:

- Tipo de onda: Seno, cuadrada, diente de sierra o triangular.
- Frecuencia de la señal a representar.
- Fase
- Offset
- Frecuencia de muestreo
- Imagen de la señal

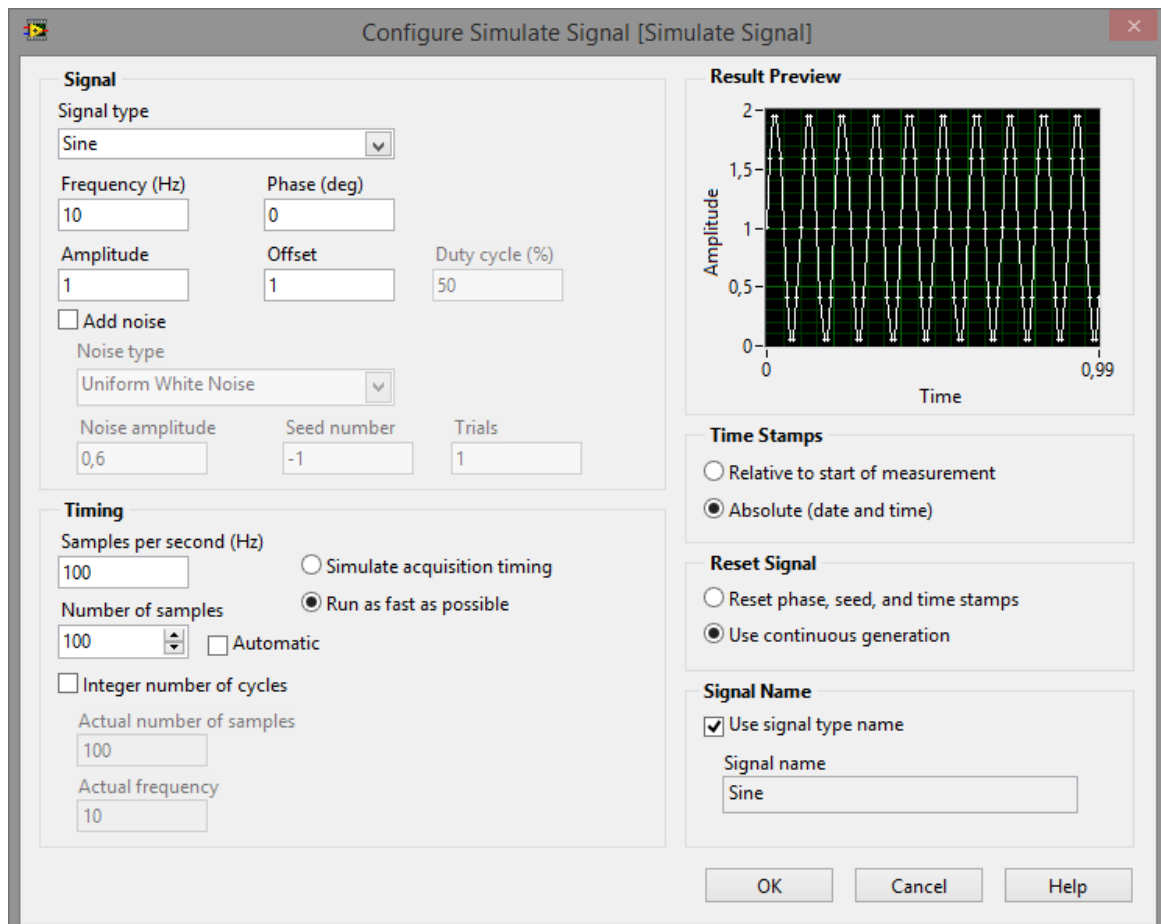


Figura 21

En nuestro caso, fijamos las siguientes características del programa:

- Tipo de Señal: Seno
- Frecuencia: 10 Hz
- Fase: 0
- Amplitud: 1V
- Offset: 1V
- Frecuencia de muestreo: 100 muestras/s

Algunos ejemplos característicos:

- 1) Señal de 20Hz, duración 1s

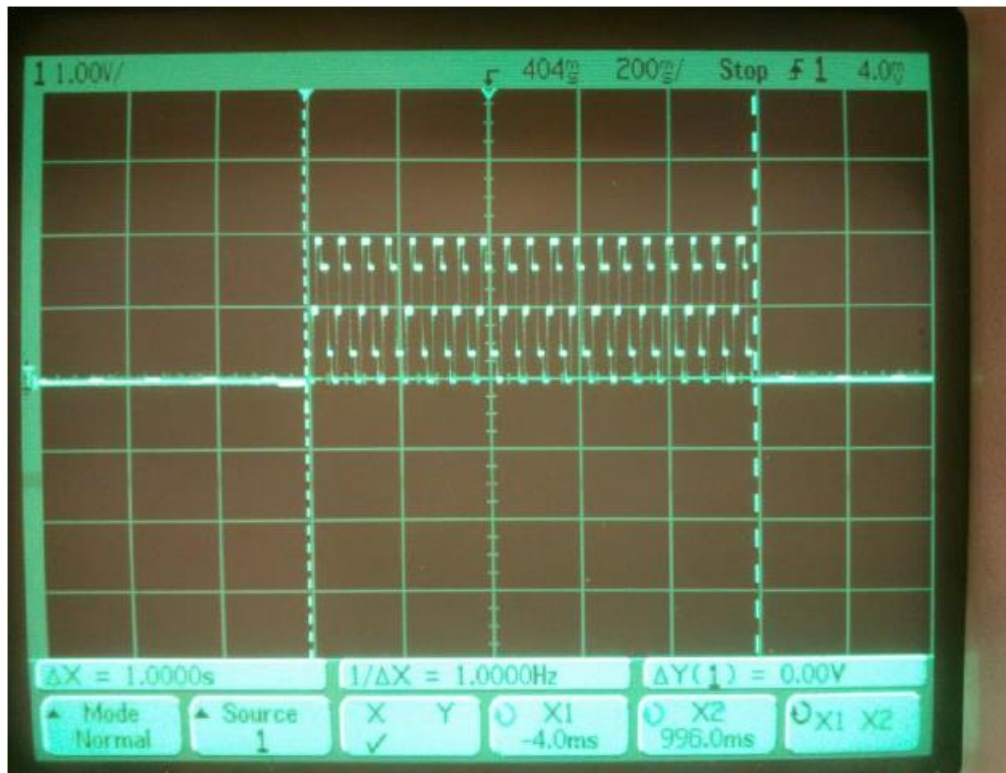


Figura 22

- 2) Señal de 15Hz, duración 1s

2.1) Fichero de muestras capturadas de una señal de 15 Hz a una frecuencia de 100 muestras/s

escribe: Bloc de notas										
Archivo	Edición	Formato	Ver	Ayuda						
1,000	1,951	1,588	0,412	0,049	1,000	1,951	1,588	0,412	0,049	1,000

Figura 23

2.2) Señal representada en el osciloscopio donde vemos el valor de las muestras representadas anteriormente.



Figura 24



Figura 25

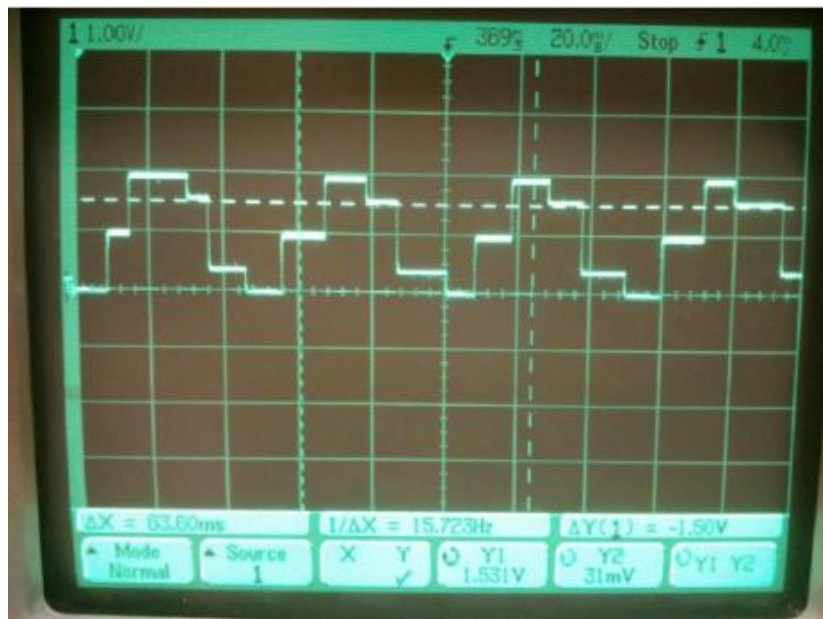


Figura 26

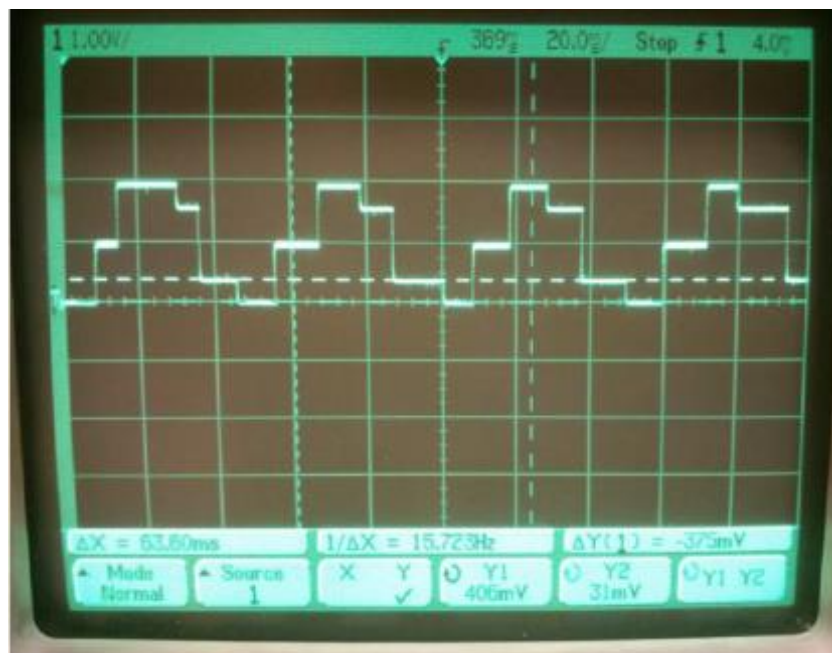


Figura 27



Figura 28

3) Señal de 30Hz, duración 1s

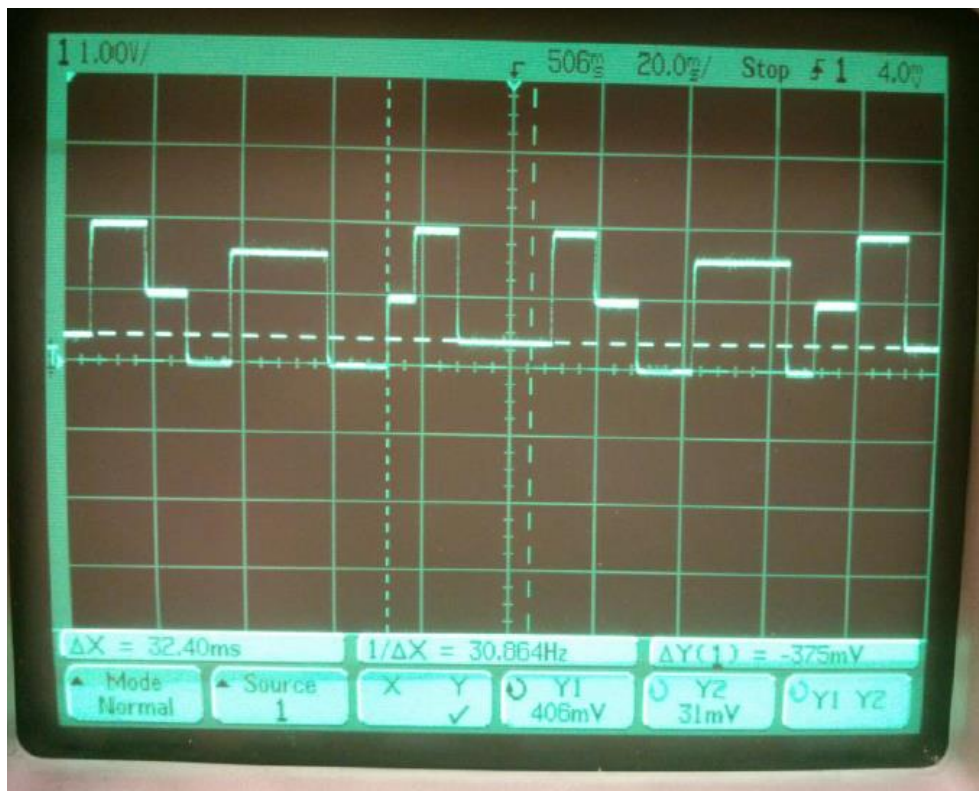


Figura 29

Como hemos visto y explicado anteriormente, una señal de 30Hz, muestreada a 100Hz, tenemos las muestras que observamos en el osciloscopio, dando una señal que no se corresponde con una senoidal completa.

Conclusiones

Para poder analizar si las pruebas han sido correctas, tenemos que analizar el motivo por el cual hemos utilizado dicho software.

En primer lugar, recordamos que el principal objetivo de esta parte es la simulación de una señal, la cual seamos capaces de sacar por la fuente de alimentación.

Como hemos podido observar en las pruebas, la única limitación impuesta por la fuente de alimentación es que los valores que queramos representar no deberán variar en al menos 10ms. Dicha limitación hace que sólo podamos sacar señales que no pasen de los 50 Hz – onda cuadrada- y 30 Hz –onda senoidal-.

En principio, estos valores son aceptables para la representación de las señales que queremos simular. No necesitamos simular señales con una frecuencia alta. Nuestro principal interés es saber si lo que está escrito en el fichero donde representamos la señal puede salir con exactitud por la fuente de alimentación.

Ahora que somos capaces de representar una señal en la fuente de alimentación, podremos pasar a la siguiente parte del proyecto donde veremos una serie de procesos que sufrirá la señal para poder ser tratada por el procesador de un sistema empujado.

Parte 2 – Hardware

Convertidor V/I

Introducción

Como hemos visto en el preámbulo, necesitamos de un conversor de Tensión/Corriente, que se encargará de convertir la tensión recibida de la señal creada en la fuente de alimentación y la transformará en corriente para pasarlo al ADC de la placa del apartado siguiente - ADC Board - y así pasarle la información del consumo a la Beagle Board.

Para la creación de un convertidor tensión-corriente, hemos estudiado el apartado de la electrónica que trata los amplificadores. Este tipo de convertidor puede tratarse de un amplificador de transconductancia, donde la corriente de salida está en función de la tensión de entrada.

Para llevar a cabo este convertidor barajamos dos tipos de circuitos:

- Convertidor con carga referenciada
- Convertidor con carga flotante

Cualquiera de estos dos tipos de convertidores podría servirnos, pero dado que este convertidor tendrá que unirse a la placa del Proyecto ADA -ADC Board-, tendremos que fijarnos en dicho componente para confirmar el circuito a utilizar.

La placa ADC Board se compone de 16 canales, donde cada uno de ellos hace referencia a un dominio de tensión de un sistema empotrado. Por ejemplo, nos encontramos con dominios como: Wifi, LAN, Usbs, Uarts, IO, etc. Cada uno de estos canales, se componen de un amplificador de instrumentación y un ADC que convertirá los datos recibidos para enviárselos a la Beagle Board.

Para ver de una manera gráfica dicha inserción, hemos realizado una simulación con el software PsPice, así conseguiremos hacernos una idea de cómo solucionar nuestro problema.

En la Figura 30 de la siguiente página, veremos la primera etapa de la placa ADC Board y marcaremos en rojo dónde quedaría ubicado nuestro convertidor. De los 16 canales de los que dispone esta placa, hemos escogido de ejemplo el canal que simularía la corriente consumida por una conexión de área local (RLan). Veremos cómo la tensión en bornes de esta resistencia ayudará a definir la corriente que la atraviesa mediante un amplificador de instrumentación. En definitiva, lo que queremos de nuestro convertidor es que actúe como una fuente de corriente variable, por lo que la dibujaremos como tal para más tarde explicar cuál será el circuito encargado de realizar dicha función.

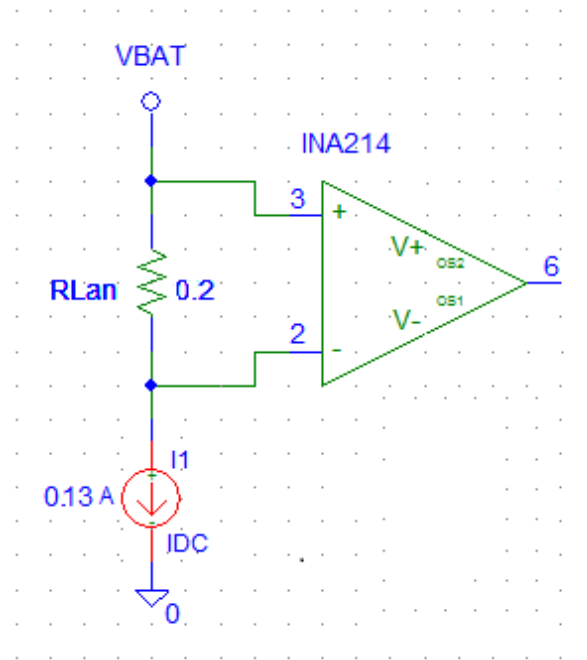


Figura 30

Convertidor V/I

Para este apartado, tenemos que diseñar un convertidor tensión-corriente. Un convertidor de este tipo, se les denomina “Amplificador de transconductancia” [7]. Estos circuitos se basan en que la corriente de salida tiene que ser en función de la tensión de entrada, pero independiente de la carga.

Existen 2 opciones, con carga flotante o con carga referenciada a masa.

Para poder decidirnos por uno de ellos, tenemos que fijarnos en cómo podemos unirlo a la tarjeta diseñada para el Proyecto ADA.

La placa ADC Board, simula las corrientes consumidas por cada uno de los dominios de tensión gracias a una Resistencia que cierra el circuito. Nosotros queremos dar un paso más, en lugar de usar una resistencia, queremos dotar a la placa de una fuente de corriente variable, consiguiendo así simular dichos dominios sin necesidad del cambio de resistencia cada vez que queramos variar la corriente.

Una vez visto el esquemático de la placa, hemos realizado una simulación en PSpice para poder hacernos una idea de una forma gráfica.

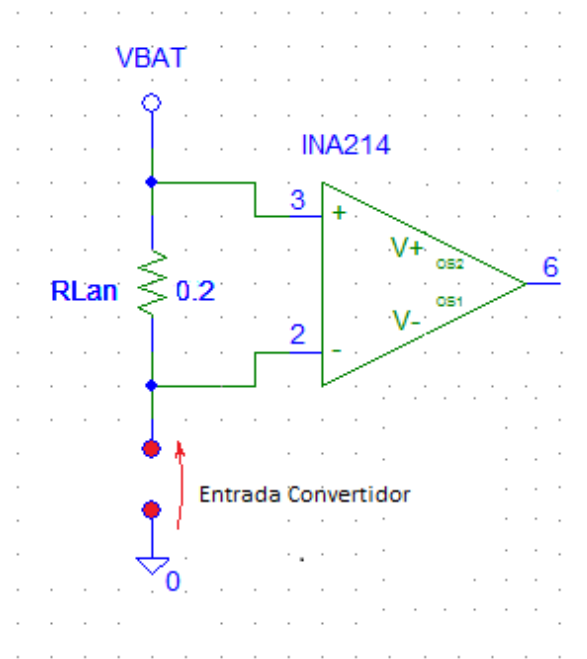


Figura 31

Estudio teórico

Una vez estudiado el esquemático de la ADC Board, vemos que es mucho más fácil el uso del convertidor referenciado a masa, también llamado “Circuito de Howland”. [7][8][9]

Con este circuito, fijándonos en las resistencias que vamos a utilizar, podemos encontrar una fácil relación V/I . La elección de dicho circuito está basada en su sencillez, hemos escogido un amplificador operacional $\mu A741$, el cual nos hemos familiarizado durante la carrera, y unas resistencias colocadas según el circuito.

Nos ayudamos del PSpice para poder realizar las simulaciones pertinentes y ver cómo quedará nuestro circuito.

Un circuito de Howland es como el siguiente:

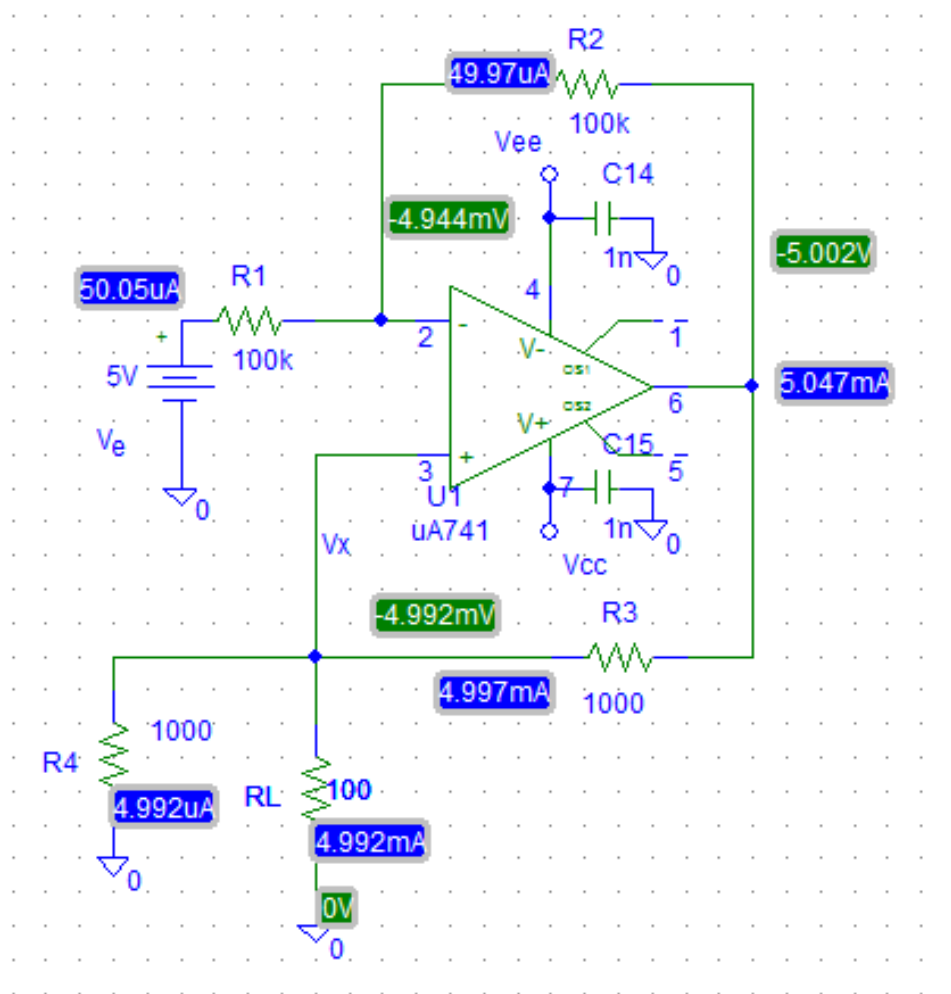


Figura 32

En él vemos que si usamos la siguiente igualdad: $R_1 = R_2$ y $R_3 = R_4$ la relación tensión corriente queda de la siguiente forma:

$$-V_E = R_4 \cdot I_L$$

Podemos ver una simulación en PSpice del circuito yendo al Anexo B.

Placa de prototipado

Para la realización de una PCB, lo primero que vamos a hacer es montar el circuito en una placa de prototipado y comprobar su funcionamiento.

El montaje sobre la placa lo hemos realizado con el esquema simulado en PSpice. En él, hemos utilizado el amplificador de instrumentación uA741 de Texas Instrument. Este amplificador se alimenta con una fuente de alimentación a 18 V y cuenta con una protección interna de cortocircuito.

En el apartado de *Resultados* veremos las pruebas realizadas para poder realizar la fabricación de nuestra PCB.

Realización PCB

Una vez verificado el circuito procedemos a su montaje en una PCB.

Para esto, elegimos la herramienta CAD con la que estemos más familiarizados. En mi caso he elegido KiCad, ya que es una herramienta gratuita que puede descargarse para cualquier plataforma y nos da unos resultados más que suficientes para nuestro proyecto.[10]

Una vez elegido el software con el que vamos a realizar la placa, lo primero es conocer el entorno para familiarizarnos con las diferentes posibilidades que nos ofrece dicho software.

Lo primero que tenemos que hacer, es una captura del esquemático completo del circuito a representar. Poniendo especial atención a los componentes seleccionados y a su manera de conectarse.

En la siguiente figura vemos una captura del esquema completo en KiCad.

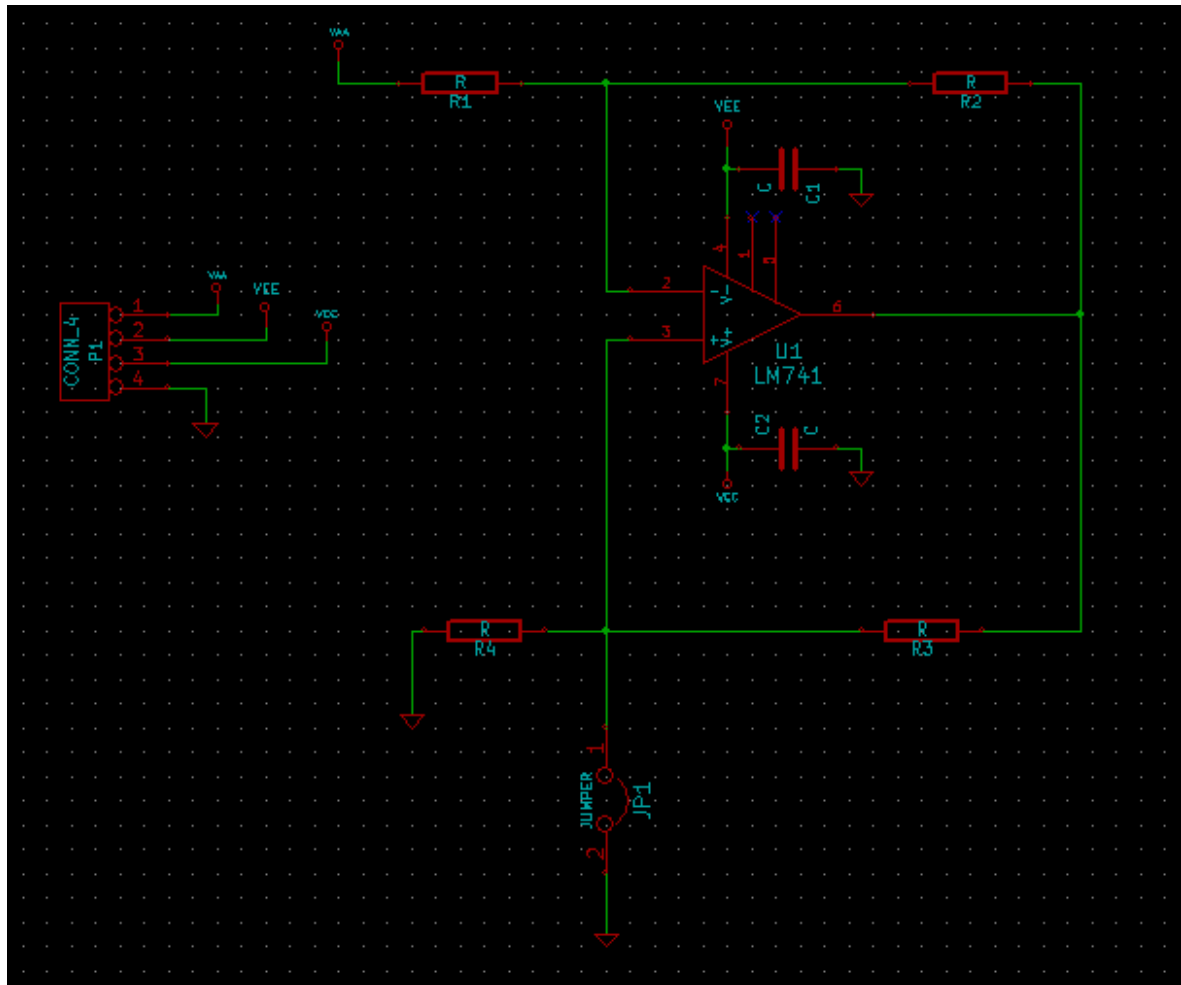


Figura 33

Una vez capturado, procedemos a elegir la huella de cada uno de los componentes que necesitemos y empezaremos con el rutado.

En la siguiente captura podremos ver cómo hemos elegido las huellas de cada uno de los componentes, usando en cada una de ellas el componente que necesitemos y verificando que su medida es la correcta.

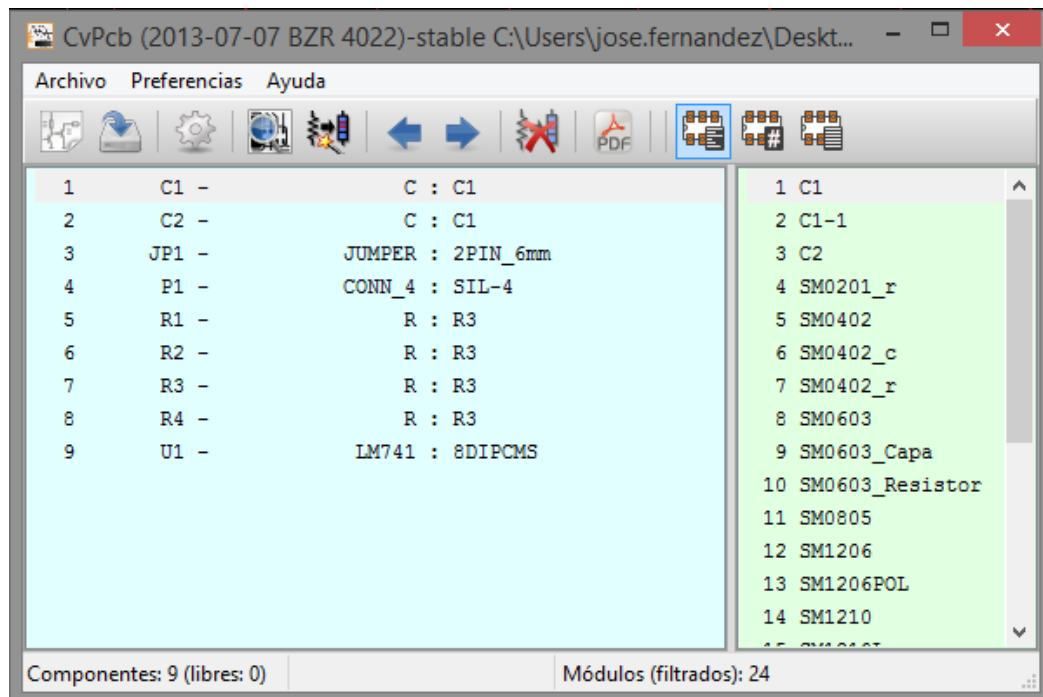


Figura 34

Una vez elegidas las huellas correctas, procedemos al rutado de la placa. Para esto, es importante colocar los componentes de forma que se minimicen los cruces de pistas y que los componentes sigan un orden lógico para su fácil comprensión.

Dada la sencillez de nuestra placa, hemos decidido colocar un plano de masa para minimizar dichos cruces y ayudarnos a una mejor colocación de los componentes.

En la Figura 33 de la página siguiente, veremos el resultado de la placa una vez completa, con sus correspondientes nombres que nos ayudaran a identificar cada uno de los componentes.

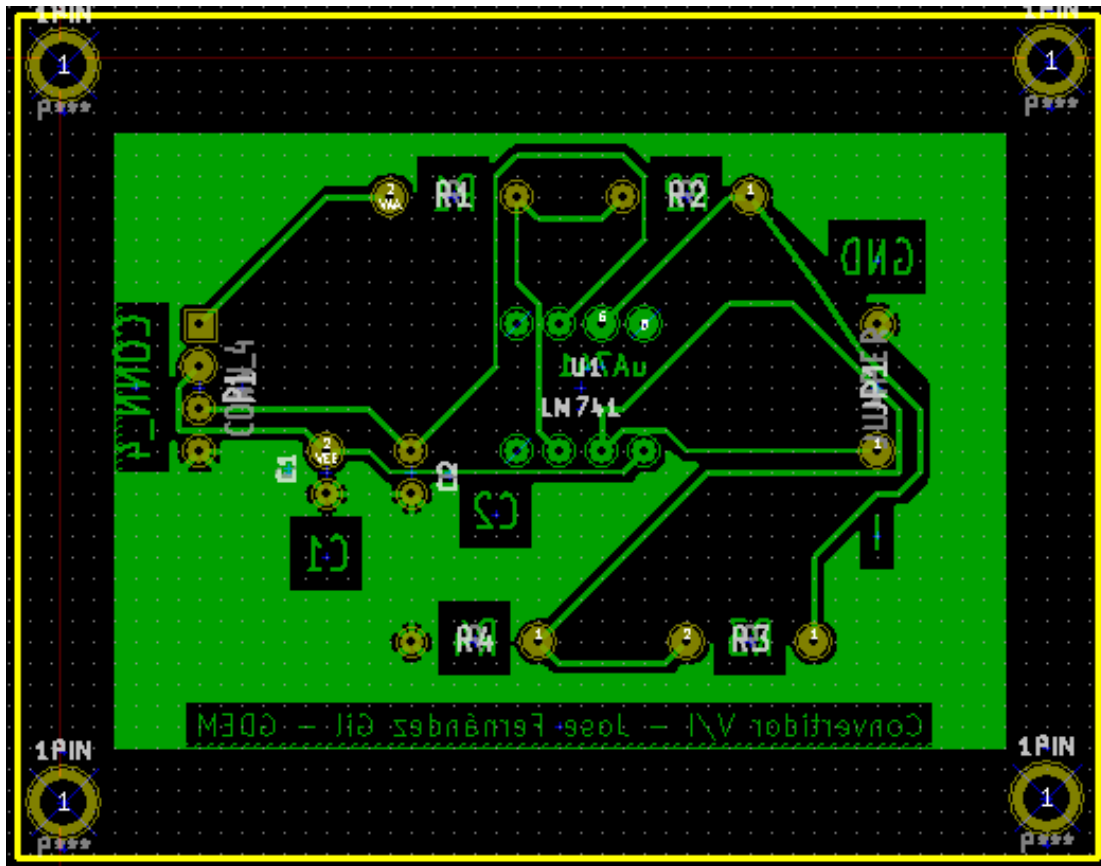


Figura 35

Finalmente, realizaremos la impresión del circuito y procederemos a su fabricación.

Para fabricar dicha placa, hemos elegido una placa fotosensible positiva, que insolaremos, para luego darle una mano de atacado, revelado y aclarado. De esta manera, podremos retirar el cobre sobrante de la placa de circuito impreso. [11]

Es importante tener en cuenta los tiempos de atacado y revelado para que la placa no sufra cortes en las pistas ni se oxide al poco tiempo. Después del aclarado le daremos una mano de alcohol que nos ayudará a quitar impurezas y hará de nuestro cobre un componente brillante y difícil de oxidar.

Una vez terminado el proceso, hemos procedido a colocar los componentes y soldarlos, como aparecen en las figuras 34 y 35, donde podemos apreciar los componentes de alimentaciones a la parte izquierda, abajo las resistencias variables y a la parte derecha la salida de nuestro convertidor que conectaremos a la placa ADC Board:

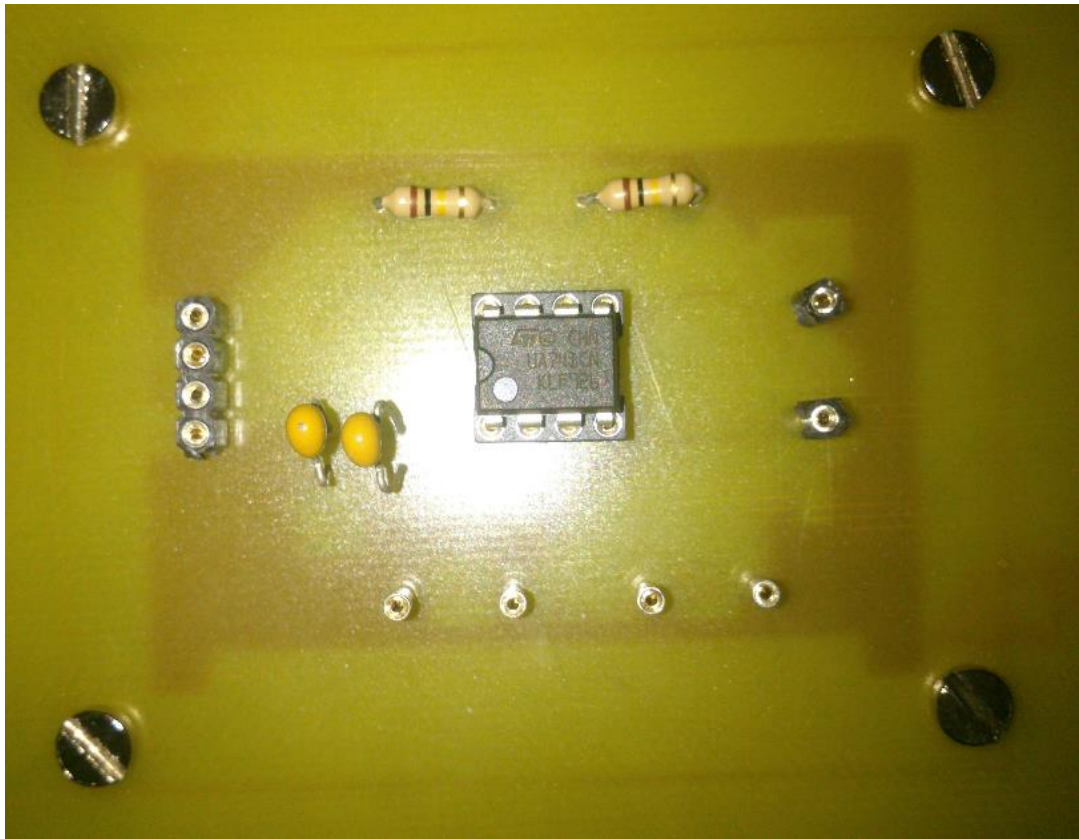


Figura 36

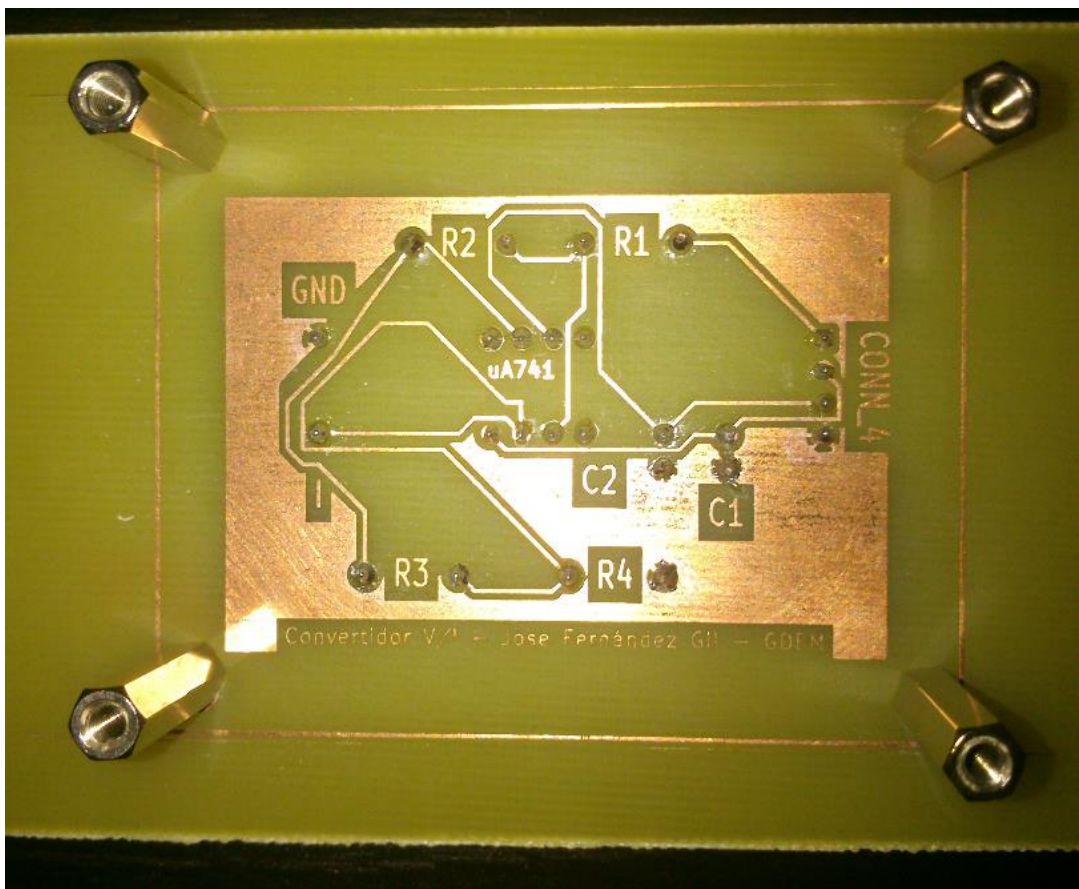


Figura 37

Como anotación, queremos destacar que hemos dejado los conectores de las resistencias R_3 y R_4 “abiertos” para poder variar la relación V/I de una manera sencilla, simplemente colocando un potenciómetro o las resistencias que le convengan, ya que para cambiar dicha relación habría que cambiar las resistencias que soldadas sería un trabajo bastante tedioso, recordando que fórmula sólo se hará efectiva si se cumple: $R_1 = R_2$ y $R_3 = R_4$

ADC Board

Introducción

La ADC Board es una tarjeta creada en el Proyecto ADA. Dicha tarjeta, se hizo como solución para crear un sistema que fuera capaz de simular los dominios de tensión de los que se compone un sistema empotrado.

Aquí nos vamos a fijar en el hardware desarrollado para poder comunicar las señales enviadas desde la fuente de alimentación a la Beagle Board.

La ADC Board consta 16 canales conectados a un amplificador de instrumentación, cuya salida va directamente a 16 entradas de un ADC, que pasaran la información contenida en ellos –valores de tensión– a través del bus SPI, uno de los buses que podemos encontrar en una Beagle Board.

El funcionamiento es el siguiente: Por cada uno de los 16 canales, tienes la opción de meter una resistencia que cerrará el circuito diseñado dando una corriente que saldrá por el amplificador de instrumentación para pasar al ADC, que realizará la conversión analógico-digital y pasará dicha información a la Beagle Board. Para pasar la información, hay que llevar a cabo una serie de configuraciones, como la activación del bus SPI y modo de funcionamiento del ADC. Estas configuraciones ya han sido desarrolladas en el Proyecto ADA. En el siguiente apartado, veremos cómo son y explicaremos cómo pueden ayudarnos a desarrollar nuestro proyecto.

Conexión ADC

Para la medida de la corriente, el circuito hace que, a la entrada diferencial del amplificador de instrumentación, se fijen 25mV, lo que al cerrar el circuito con la resistencia a simular, hace que dicha variación se convierta en la corriente solicitada.

Un ejemplo del circuito es:

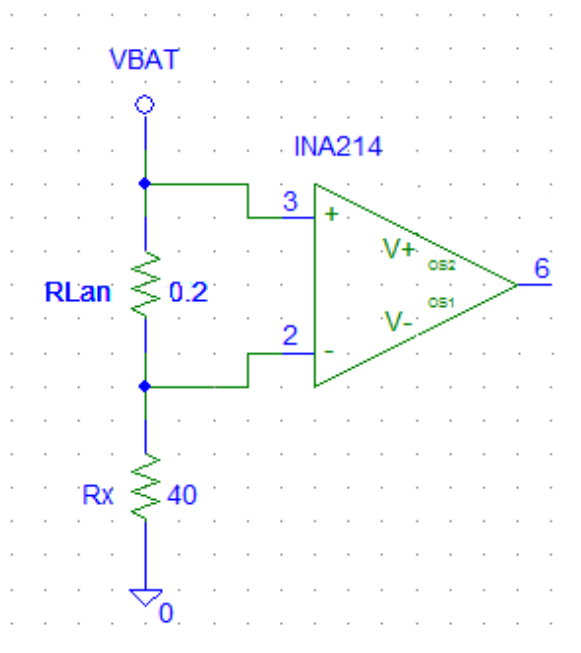


Figura 38

$$I_{MAX} = \frac{V_3 - V_2}{R_{LAN}} = \frac{25mV}{0.2}$$

$$I_{MAX} = 0.13A$$

Por lo tanto, para conseguir dicha corriente:

$$R_X = \frac{V_{BAT} - R_{LAN} \cdot I_{MAX}}{I_{MAX}}$$

$$R_X = 40\Omega$$

Este ejemplo se aplica en todos los canales como podemos ver en el Anexo C.1 para poder simular los valores máximos de corriente de los canales.

Estos valores de corriente, es la información que mandamos al ADC para podernos comunicar con la Beagle ayudándonos del siguiente esquema de conexiones [12]:

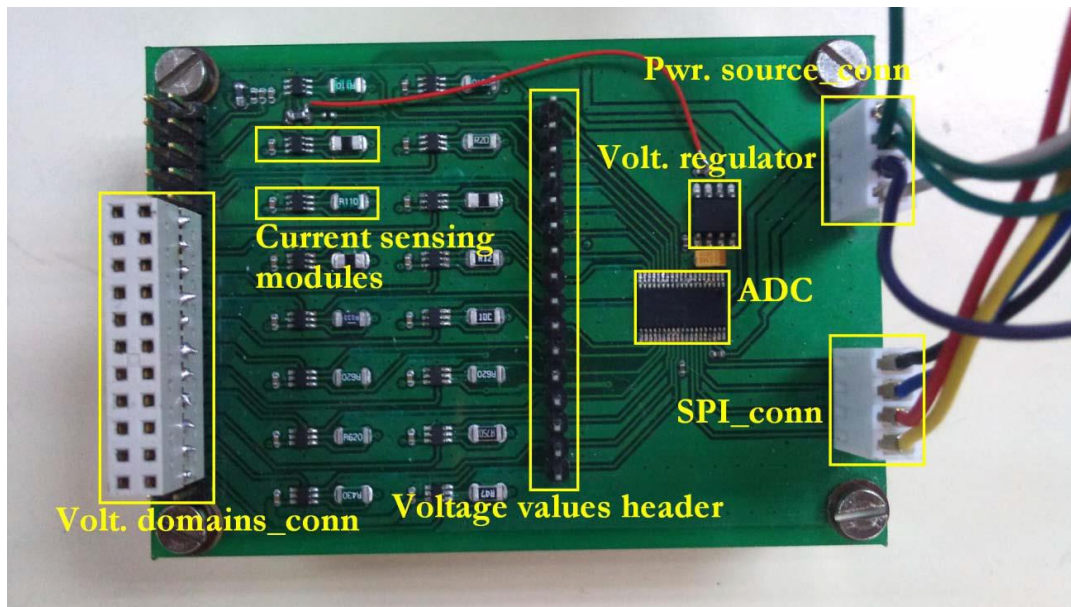


Figura 39

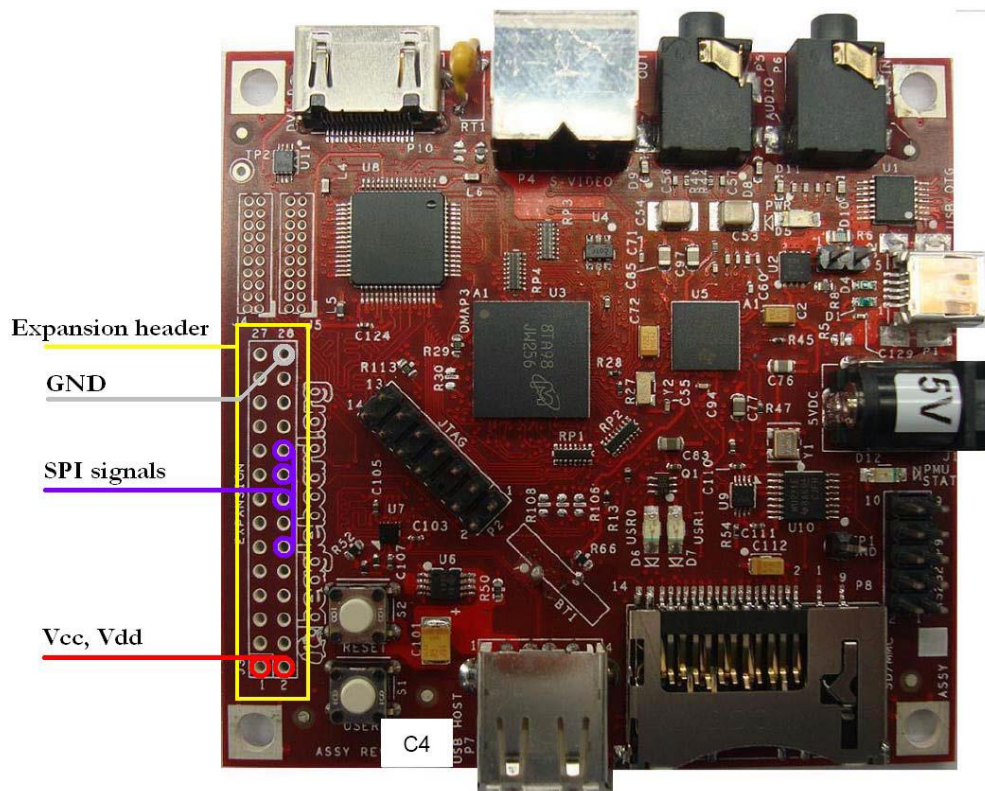


Figura 40

Como variación, se va a realizar el cambio de la resistencia conectada a “Volt. Domains_com” en la Figura 39 por una tarjeta de diseño propio –explicada en el apartado anterior- ya que nuestro interés es poder simular una señal completa con la mayor calidad posible y variación según demande el testeo.

Como hemos visto en el apartado anterior, esta tarjeta será un convertidor de tensión/corriente que entregará directamente la información de la corriente al amplificador de instrumentación, pudiendo generar una corriente variable de cualquier frecuencia.

Para hacernos una idea de la conexión de esta parte con el apartado anterior, recreamos una imagen de nuestra mesa de trabajo. En la Figura 41, podemos ver la conexión de la placa con la Beagle Board. Aquí tenemos una Beagle ya configurada y preparada para comunicarse a través del cable serie con el ordenador, aunque esto lo veremos en profundidad en el siguiente apartado.

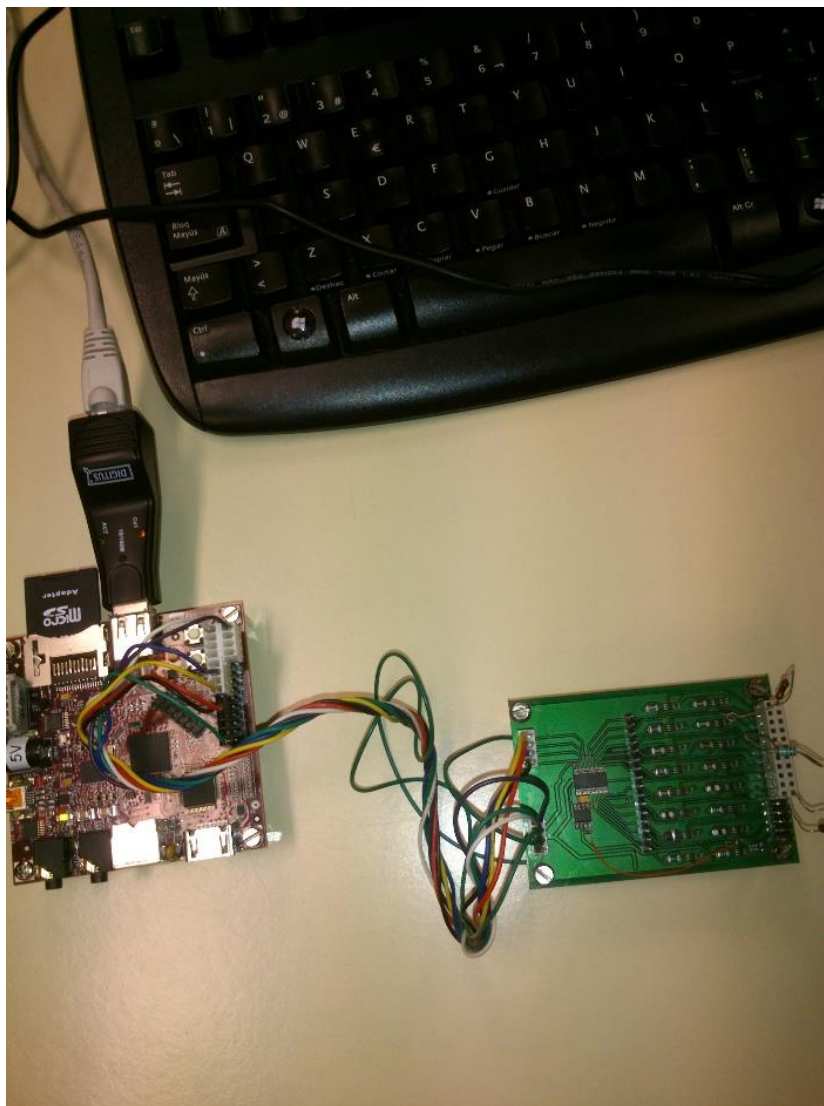


Figura 41

En la figura 41 vemos la conexión a la Beagle que, como hemos adelantado antes, se hace a través del bus SPI.

Como breve introducción al bus SPI, diremos que es un protocolo serie de comunicación, que tiene como ventajas una comunicación Full Duplex y una mayor velocidad de transmisión que con I2C o SMBus. Podremos ver una definición más completa en el apartado “Beagle Board” donde explicaremos su interconexión con la ADC Board usando dicho bus.

Para poder completar las pruebas de este apartado, necesitamos conocer cómo se conecta la Beagle al ordenador y el programa con el que poder comunicarse, por este motivo, no tiene sentido todavía realizar las pruebas de este sistema. Puesto que depende íntegramente de la Beagle, veremos sus pruebas junto a este apartado.

Para ver el esquemático de esta tarjeta, podemos ir al Anexo C.1. En éste podemos ver los componentes que han utilizado, como por ejemplo un amplificador de instrumentación INA 214. Este amplificador se trata de uno de la serie de Texas Instrument INA2XX donde el fabricante nos da la relación de ganancia que vemos en la siguiente tabla:

PRODUCT	GAIN	R_3 and R_4	R_1 and R_2
INA210	200	5 k Ω	1 M Ω
INA211	500	2 k Ω	1 M Ω
INA212	1000	1 k Ω	1 M Ω
INA213	50	20 k Ω	1 M Ω
INA214	100	10 k Ω	1 M Ω
INA215	75	13.3 k Ω	1 M Ω

También podemos ver un ADC de Texas Instrument de la serie ADS79XX – AS7953 –, un conversor multicanal de 12/10/8 bits de resolución. Unas de sus características destacadas son:

- Varios modos de configuración automáticos y manuales
- 20 Mhz interfaz serie
- 1Mhz de muestras para dispositivos serie
- 2 niveles de alarma programables por canal
- 4 GPIOs configurables individualmente
- Ancho de banda de 47 a 3dB en la entrada

Beagle Board

Introducción

La Beagle Board es un ordenador de bajo consumo integrado en una única placa, producido por Texas Instrument. Su diseño es en código abierto, tanto software como hardware, y usa un procesador OMAP3530 –SoC-, que más tarde explicaremos algunas de sus características. [14]

Esta placa, fue diseñada por un pequeño equipo de ingenieros que querían darle un uso didáctico en universidades para poder extender los conocimientos de lo que hoy en día se conoce como “Open Source”. La placa se construyó usando Orcad para los esquemáticos y Allegro para su fabricación. Como curiosidad, según estos ingenieros, no se utilizó ninguna herramienta de simulación.

Para poder describir las características de la Beagle Board, nos vamos a ayudar de un pequeño esquema que nos encontramos en su misma página web y que nos va a ayudar a la hora de poder comprender cómo ha logrado este tipo de tarjetas “didácticas” situarse en el mercado.

Esquema

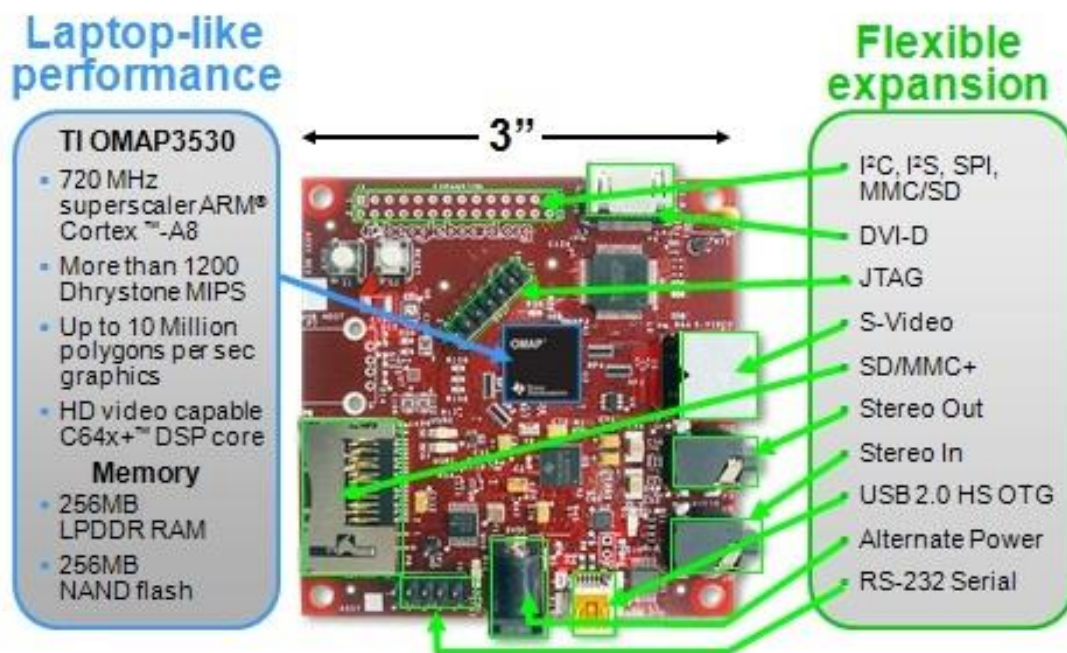


Figura 42

En el esquema podemos ver la colocación de los componentes y sus posibles alternativas para diferentes usos. En nuestro caso, dado que no se realizará ningún estudio de video ni audio, solo nos interesa la salida serie RS-232, la entrada SD/MMC+ para meter el sistema operativo y el bus SPI mediante el cual conectaremos la placa ADC Board. [15]

- Procesador:

La Beagle utiliza un procesador OMAP3530DCBB72 720MHZ. Dicho procesador, utiliza la tecnología de montar la memoria NAND y SRAM en la parte de arriba del procesador (POP, que significa, Package on Package).

- Memoria:

La tecnología POP proporciona las siguientes características:

- 2Gb NAND x 16 (256MB)
- 2Gb MDDR SDRAM x32 (256MB @ 166MHz)

En la placa no podemos encontrar más memorias, pero es posible agregarlas mediante métodos extraíbles:

- Slot para una tarjeta SD/MMC
- Slot para USB OTG
- Slot para USB EHCI

- RS232:

Es proporcionado por un conector de 10 pines, que requiere de un conector IDC o DB9 para poder acceder al puerto serie del equipo.

- SD/MMC+:

Soporta tarjetas de todo tipo, Wifi, cámara, bluetooth, GPS... Nosotros usaremos una tarjeta SDHC para instalar el sistema operativo. Es importante destacar, que es necesario una tarjeta de un tamaño máximo de 8Gb y una velocidad clase 10 (10Mb/s) para poder correr en ella un sistema operativo lo más fluido posible.

- McSPI:

Una de las características del procesador OMAP3530 es el bus Multichannel SPI (McSPI). Este procesador cuenta con 4 controladores McSPI. Cada uno de estos controladores tiene un número de entradas. De esos 4, solo el McSPI3 y el McSPI4 tienen la posibilidad de salir fuera del procesador y colocarse en la zona de pines de expansión de la BeagleBoard. Para

este proyecto, trabajaremos con el 4 controlador que cuenta con entrada de Chip-Select, el McSPI4.

Cuando conectemos las placas correspondientes, volveremos a hacer referencia a estos pines y a explicar cuáles serán sus funciones.

Entorno de trabajo.

Nuestro entorno de trabajo se compondrá de una BeagleBoard, un ordenador conectado mediante un cable serie (RS232) y una tarjeta SDHC.

Como ya sabemos, para poder llevar a cabo esta tarea, nos hemos ayudado del Proyecto ADA. [12] Este proyecto, conecta la BeagleBoard a la ya conocida ADC Board realizada por el grupo. Sin embargo, en este apartado, tan solo nos vamos a fijar en la conexión de la Beagle al ordenador.

Gracias a la intranet del grupo GDEM [16], podemos seguir unos sencillos pasos para la instalación del sistema operativo y la primera conexión al mismo mediante un software de comunicación serie (Putty en Windows y Minicom en Linux). Dado que en nuestro proyecto, trabajamos con Labview, nos ha resultado más cómodo realizar dichas pruebas en Windows.

Para poder introducir el sistema operativo, tenemos que hacer 2 particiones en nuestra SD, una para recoger el sistema de ficheros y otra para los archivos arrancables. También tenemos la opción de montar un servidor NFS para poder compilar nuestro software sin tener que molestarnos en sacar la tarjeta SD, pero dado que sólo vamos a compilar un programa, no vemos necesario dicho montaje.

Para la creación de la tarjeta necesitamos los siguientes archivos:

- MLO + uBoot:

Se corresponden al bootloader de la tarjeta, se trata de un programa que se ejecuta automáticamente cuando se inicia el sistema y se encarga de ejecutar el sistema operativo.

En nuestro sistema utilizamos dos bootloaders, debido a que OMAP tiene un sistema de bootloaders en dos etapas: en la primera el hardware de la CPU carga, siempre, en una memoria interna de tan solo 64kB, el código del archivo llamado MLO (el programa que se conoce como X-Loader). Al ser una memoria tan limitada, el programa solo puede hacer configuraciones del sistema muy básicas y cargar un archivo mayor, en nuestro caso el uBoot. Este archivo se carga en la RAM y puede ser mucho más grande y realizar acciones más complejas. En nuestro caso, el uBoot saca una consola por el puerto serie, a través de la cual podemos realizar acciones básicas sobre el hardware, cargar archivos binarios en la RAM, ejecutarlos...

- ulmage:

El binario del kernel que podremos cargar en la Beagle gracias a los dos archivos comentados anteriormente.

Una vez creada nuestra tarjeta, procedemos a la carga del sistema operativo conectando la SD a la Beagle y conectándola al ordenador mediante un software de Windows de comunicación serie. Una vez arrancado el sistema, dado que tenemos el sistema operativo en la tarjeta, tendremos que parar el autoarranque para entrar con el siguiente control:

```
Texas Instruments X-Loader 1.4.2 (Feb 19 2009 - 12:01:24)
Reading boot sector
Loading u-boot.bin from mmc

U-Boot 2009.01 (Oct 07 2011 - 19:13:01)

I2C:   ready
OMAP3530-GP rev 2, CPU-OPP2 L3-165MHz
OMAP3 Beagle board + LPDDR/NAND
DRAM:  256 MB
NAND:  256 MiB
MUSB: using high speed
In:     serial usbttty
Out:    serial usbttty
Err:    serial usbttty
Board revision Ax/Bx
Serial #339800040000000004039f220201c007
Hit any key to stop autoboot:  0
OMAP3 beagleboard.org #
OMAP3 beagleboard.org # run mmc mch
Usage:
mmcinit - init mmc card

reading uImage
```

Figura 43

Para poder ver correctamente en el ordenador los comandos de la Beagle usaremos la siguiente configuración:

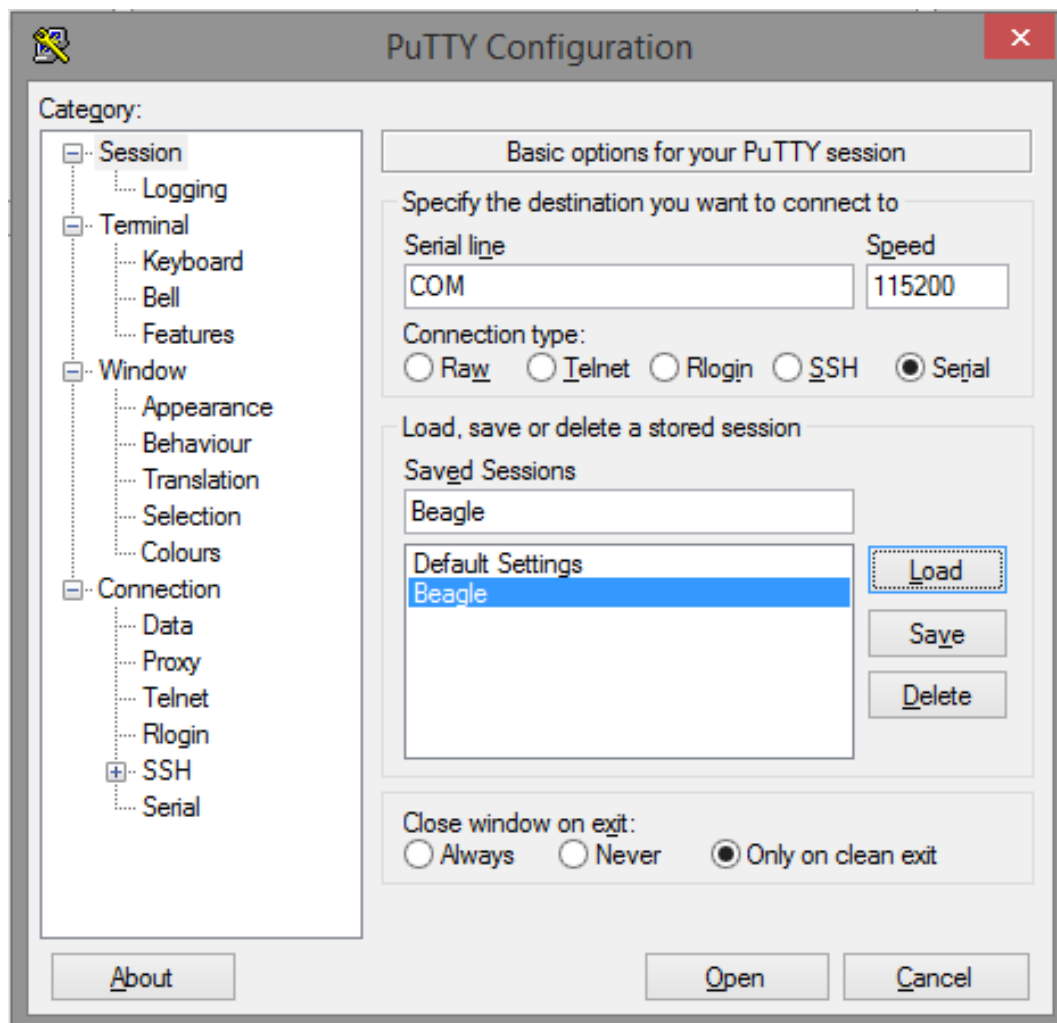


Figura 44

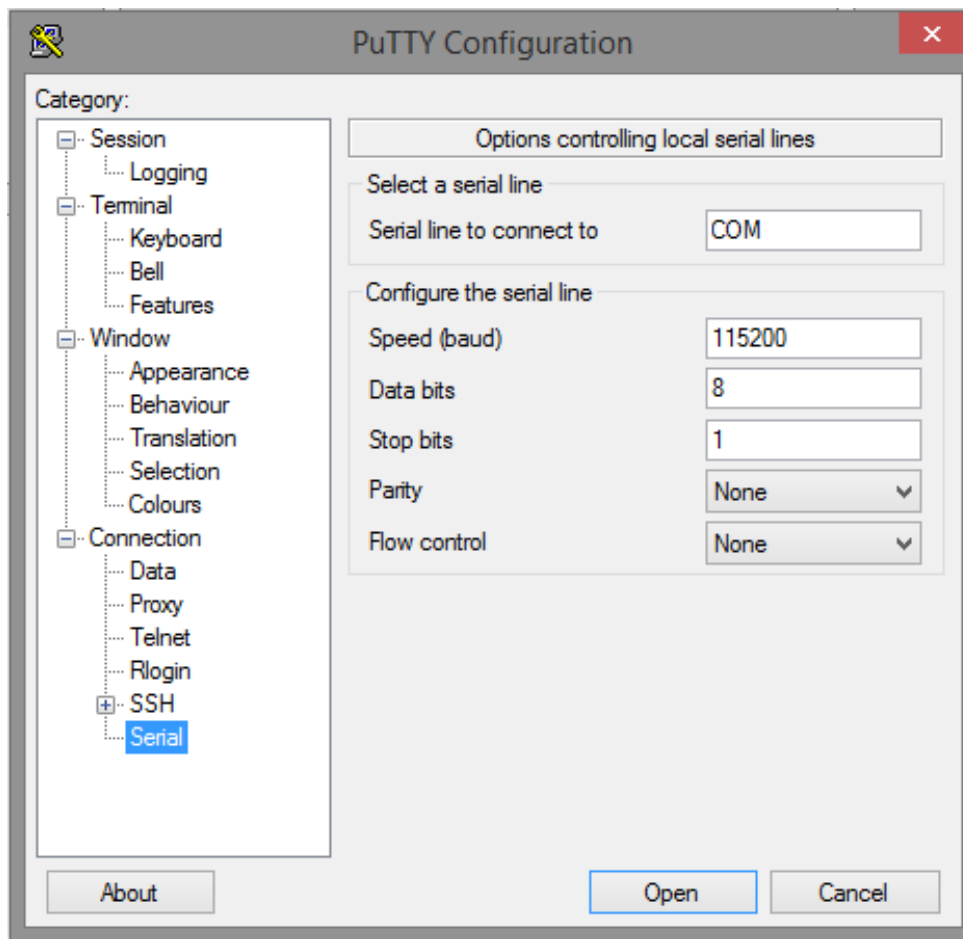


Figura 45

Vemos que la configuración de la comunicación serie son 115200 baudios, 8 bits con 1 de parada y sin paridad ni control de filas.

Hay una variación respecto a nuestro proyecto, debida al uso del Labview. En principio, este proyecto está pensado para el uso del sistema operativo Linux Ubuntu. En esas condiciones, solamente habría que variar el software serie del Putty al Minicom. En este software no es necesaria la configuración de la velocidad, ya que la coge de forma predefinida. El resto del comportamiento es similar, ya que el sistema operativo que hay instalado en la tarjeta tiene como base Linux.

Una vez arrancado el sistema, nos aparecerá una pantalla como la siguiente:

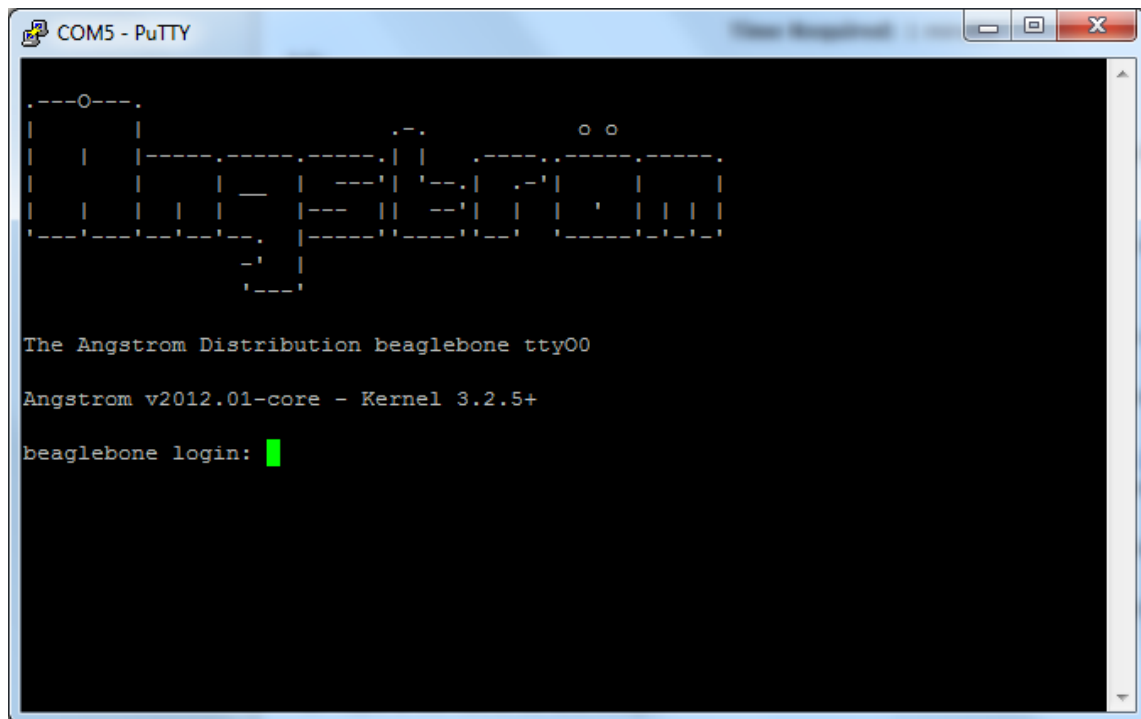


Figura 46

En esta pantalla de inicio, tenemos que ingresar con el usuario root, cuya contraseña es root. Una vez dentro, podemos movernos por el sistema operativo que hemos instalado.

Dentro, nos encontramos con el sistema de archivos básico de Linux:

- /: Raíz.
- /usr: Aquí se encuentra la gran mayoría de los archivos existentes en Linux.
- /bin: Aquí están los comandos que pueden usar todos los usuarios
- /sbin: Aquí están los comandos que sólo puede usar el administrador o root.
- /dev: Ahí están todos los dispositivos de nuestra máquina.
- /home: Lugar donde se almacenan las cuentas de usuarios.
- /lib: Enlaces a las librerías que se necesitan para el sistema.
- /var: Contiene información variable, como por ejemplo los logs del sistema
- /tmp: Directorio temporal.
- /etc: Configuración global de los programas.
- /root: Cuenta del administrador.
- /boot: Aquí está todo lo necesario para arrancar el sistema.
- /media: Punto de montaje para sistemas de archivos montados localmente.
- /mnt: Antiguo punto de montaje para sistemas de archivos montados localmente. Hoy en día está en desuso y se conserva por razones históricas.
- /proc: Sistema de archivos virtual de información de procesos y del kernel.

Tan solo nos queda la creación del directorio /opt para compilar y ejecutar los programas que hallamos creado

Programación

Una vez preparado el entorno, nos disponemos a habilitar la entrada SPI para poder comunicarnos con la tarjeta ADC board -explicada en el anterior apartado-.

El Bus SPI (del inglés Serial Peripheral Interface) es un estándar de comunicaciones, usado principalmente para la transferencia de información entre circuitos integrados en equipos electrónicos. El bus de interfaz de periféricos serie o bus SPI es un estándar para controlar casi cualquier dispositivo electrónico digital que acepte un flujo de bits serie regulado por un reloj.

Las principales características de este bus son:

- Transmisión de datos síncrona en modo Full-Duplex
- Rango de frecuencias de 1 a 70MHz
- Arquitectura Master-Slave
- Tamaño de palabra variable (8, 10, 12 o 16 bits por palabra)
- Usa la comunicación serie a través de 3 ó 4 cables (sin entrada Chip-Select)
 - o SCLK: Serial Clock
 - o MOSI (Master Output/Slave Input) o SDO (Serial Data Output, from master)
 - o MISO (Master Input/Slave Output) o SDI (Serial Data Input, toward master)
 - o /SS o /CS: Chip-Select

Nos ayudamos del datasheet de la placa BeagleBoard para conseguir la dirección exacta de este bus:

Pin	Modo 0	Modo 1	Conf. Pin
20	McBSP1_CLKR	McSPI4_CLK	OMAP_PIN_INPUT
18	McBSP1_DR	McSPI4_SOMI	OMAP_PIN_INPUT
16	McBSP1_FSX	McSPI4_CS	OMAP_PIN_OUTPUT
12	McBSP1_DX	McSPI4_SIMO	OMAP_PIN_OUTPUT

Una vez preparado el bus SPI, el Proyecto ADA se encargaba de pasar por dicho bus la información que le pasa la ADC Board y la muestra por pantalla. Para llevar a cabo este procedimiento, en el Proyecto, se elaboró un programa en C que contenía la configuración de dicho bus y el modo de funcionamiento del ADC.

Veamos un ejemplo del programa original para familiarizarnos con el entorno, aunque las pruebas completas las veremos en el siguiente apartado de *Resultados*.


```

root@beagleboard:/# ./testbeagle
spi mode: 0
bits per word: 8
max speed: 50000 Hz (50 KHz)
number of channels: 16
ADC automatic mode 1 has been configured

TPS      Consumption:    0.138986 mA 0.736627 mW
MPU_IVA Consumption:    0.000000 mA 0.000000 mW
CORE     Consumption:   25.104603 mA 133.054398 mW
IO        Consumption:    0.000000 mA 0.000000 mW
SRAM      Consumption:    0.000000 mA 0.000000 mW
MEM       Consumption:    0.000000 mA 0.000000 mW
UARTS     Consumption:    0.000000 mA 0.000000 mW
USBS      Consumption:   14.966516 mA 79.322540 mW
CAM       Consumption:    0.122100 mA 0.647131 mW
DVI       Consumption:   24.069883 mA 127.570389 mW
SCREEN    Consumption:    0.000000 mA 0.000000 mW
MMC1      Consumption:   69.580429 mA 368.776276 mW
LAN       Consumption:    0.000000 mA 0.000000 mW
WIFI      Consumption:    0.000000 mA 0.000000 mW
IMON      Consumption:   47.435898 mA 251.410263 mW
OTHER     Consumption:    0.000000 mA 0.000000 mW

Elapsed time: 0 sec ::164459 usec

```

Figura 47

Explicación:

El programa del Proyecto ADA, se basa en la toma de diez muestras de cada uno de los dominios de tensión comentados anteriormente. Estas muestras se van almacenando en un buffer para realizar una media y mostrarla por pantalla como vemos en la figura 47. Donde solamente tendrán consumo los canales en los que hemos insertado una resistencia para simular el paso de corriente a través de ellos.

Para nuestro proyecto, la medida que muestra por pantalla, no nos ayuda a saber si se le está pasando una señal completa, por lo que hemos realizado una serie de modificaciones para que podamos ver por pantalla todas las muestras tomadas y así hacernos una idea de si le pasa al procesador la señal completa enviada desde la fuente de alimentación Agilent.

La modificación que queremos realizar es que el programa no realice la media de las muestras, si no que muestre todas por pantalla. Para ello, simplemente, comentamos las líneas de código donde se realiza y modificamos las variables a mostrar. También se puede modificar el número de muestras, simplemente modificando la variable global `#num_samples`. Aunque nosotros hayamos decidido dejarla como está, 10 muestras para ver lo que le está pasando al proceso.

A continuación vemos una parte del código del programa del Proyecto ADA [12] donde veremos las modificaciones, marcadas como comentario:

```
void calc_operations (void)
{
    int k=0, i=0;

    for (k=0; k < num_channels;k++)
    {
        for (i = 0; i < num_samples; i++)
        {
            dominios[chs_read[k]].measures[i].measure =
                25000*dominios[chs_read[k]].measures[i].measure;

            dominios[chs_read[k]].measures[i].measure =
                dominios[chs_read[k]].measures[i].measure/4095;

            dominios[chs_read[k]].measures[i].measure =
                dominios[chs_read[k]].measures[i].measure/dominios[chs_read[k]].
                Rsense;

            //dominios[chs_read[k]].avg_measure=dominios[chs_read[k]].avg_measure +
            //dominios[chs_read[k]].measures[i].measure;
            //Borramos la linea de código correspondiente a la media

            if (i==0)
                printf("%s\tConsumption:\n",dominios[chs_read[k]].dominio);
            //Mostramos las 10 medidas con solo un enunciado

            if ((k==3) || (k==4) || (k==5) || (k==6) || (k==8) || (k==10) || (k==11) || (k==12))
                printf("\t%f mA %f mW \n", dominios[chs_read[k]].measures[i].measure,
                    (dominios[chs_read[k]].voltage)*(dominios[chs_read[k]].measures[i].meas
                    ure));

        }
    }
}
```

Compilamos dicho programa gracias a la instalación del compilador *gcc* en la Beagle y podemos obtener unas primeras pruebas del sistema. A continuación realizamos una prueba aislada. Para una mayor comprensión del sistema completo veremos más pruebas en el siguiente apartado.

Sistema Completo:

```
root@beagleboard:/miopt# gcc miAPIsp2i.c -o testi
root@beagleboard:/miopt# ./testi
spi mode: 0
bits per word: 8
max speed: 50000 Hz (50 KHz)
number of channels: 16
ADC automatic mode 1 has been configured
```

IO	Consumption:
	5.268029 mA 27.920555 mW
	5.140021 mA 27.242113 mW
	5.100634 mA 27.033361 mW
	5.218796 mA 27.659618 mW
	5.573279 mA 29.538382 mW
	4.943086 mA 26.198355 mW
	5.277876 mA 27.972744 mW
	5.159715 mA 27.346489 mW
	5.445271 mA 28.859940 mW
	5.218796 mA 27.659618 mW
SRAM	Consumption:
	0.000000 mA 0.000000 mW
	0.000000 mA 0.000000 mW
	0.000000 mA 0.000000 mW
	0.000000 mA 0.000000 mW
	0.000000 mA 0.000000 mW
	0.000000 mA 0.000000 mW
	0.000000 mA 0.000000 mW
	0.000000 mA 0.000000 mW
	0.000000 mA 0.000000 mW
	0.000000 mA 0.000000 mW

Figura 48

Resultados – Parte 2

Como hemos realizado en el apartado anterior. Ya hemos explicado de las partes de las que se compone este segundo apartado. Por lo que vamos a realizar las pruebas pertinentes para verificar su comportamiento.

Para realizar estas pruebas, vamos a volver a dividir este mismo bloque en dos partes. Una primera parte será la prueba del sistema Beagle + ADC Board y una segunda parte uniendo a este sistema el convertidor V/I.

Para verificar un correcto funcionamiento debemos comprobar varios escenarios.

Primero, tenemos que verificar que la Beagle y el ADC son capaces de sacar los valores que esperamos. Por ejemplo, si seguimos tabla de resistencias que hemos adjuntado en el Anexo C.2, veremos que si metemos una resistencia de 100Ω al canal 2, tiene que salirnos por pantalla que la corriente del sistema son 50mA.

Una vez verificado que los valores entran dentro de la normalidad podemos pasar al siguiente caso.

Segundo, deberemos comprobar si el dato se mantiene estable sin variar el valor de resistencia en ninguno de los canales. Con esto descartaremos posibles problemas en el código y posibles problemas tanto en el amplificador de instrumentación como en el ADC.

Dichos valores están pensados para unos valores máximos de corriente y están protegidos con un regulador para evitar sobretensión.

Una vez realizadas varias pruebas y verificado que el comportamiento es como esperamos, podemos dar un siguiente paso, la conexión del convertidor.

Para poder probar el sistema completo, vamos a hacer uso de la fuente de alimentación – sin usar el Labview- Simplemente fijamos una tensión continua y comprobamos mediante el código final modificado si es capaz de retener dicho valor. Es importante conectar las 3 tarjetas poniendo especial atención a las alimentaciones y las masas.

También, es importante destacar que nos faltan todavía unas últimas pruebas verificando el sistema completo. Por lo tanto, todavía parecerán unas pruebas incompletas.

Beagle + ADC Board

Partimos de la base de haber conectado la Beagle al PC y ver la comunicación serie mediante el software Putty. Y partimos también de haber sido capaces de conectar correctamente la tarjeta ADC Board con la Beagle.

Una vez en la línea de comandos del Putty y después de compilar y habilitar el bus SPI mediante el programa comentado en el apartado de la Beagle, podemos empezar a trabajar.

En una primera instancia, vamos a realizar la prueba con el programa inicial. Donde nos encontramos con los 16 canales mostrados por pantalla con una sola muestra. Conectamos dos resistencias a dos canales distintos y nos aparecen unos valores de consumo en dichos canales y cero en los demás.

Vemos en las siguientes figuras dichas muestras:



Figura 49


```
Tue Jan 21, 19:52:50

jfernandez@ubuntu: ~
File Edit View Terminal Help

USBS Consumption: 0.000000 mA 0.000000 mW
CAM Consumption: 0.000000 mA 0.000000 mW
DVI Consumption: 0.000000 mA 0.000000 mW
SCREEN Consumption: 0.000000 mA 0.000000 mW
MMC1 Consumption: 0.000000 mA 0.000000 mW
LAN Consumption: 0.000000 mA 0.000000 mW
WIFI Consumption: 0.000000 mA 0.000000 mW
IMON Consumption: 0.000000 mA 0.000000 mW
OTHER Consumption: 0.000000 mA 0.000000 mW

Elapsed time: -1293064401 sec ::1759399 usec
Segmentation fault
root@beagleboard:/# ./testbeagle
spi mode: 0
bits per word: 8
max speed: 50000 Hz (50 KHz)
number of channels: 16
ADC automatic mode 1 has been configured

TPS Consumption: 0.000000 mA 0.000000 mW
MPU_IVA Consumption: 57.374279 mA 304.083679 mW
CORE Consumption: 0.000000 mA 0.000000 mW
IO Consumption: 0.000000 mA 0.000000 mW
SRAM Consumption: 0.000000 mA 0.000000 mW
MEM Consumption: 40.322578 mA 72.580643 mW
UARTS Consumption: 0.000000 mA 0.000000 mW
USBS Consumption: 0.018500 mA 0.098050 mW
CAM Consumption: 0.000000 mA 0.000000 mW
DVI Consumption: 0.000000 mA 0.000000 mW
SCREEN Consumption: 0.000000 mA 0.000000 mW
MMC1 Consumption: 0.000000 mA 0.000000 mW
LAN Consumption: 0.000000 mA 0.000000 mW
WIFI Consumption: 0.000000 mA 0.000000 mW
IMON Consumption: 1.964702 mA 10.412921 mW
OTHER Consumption: 0.000000 mA 0.000000 mW

Elapsed time: 0 sec ::232177 usec
Segmentation fault
root@beagleboard:/#
CTRL-A Z for help | 115200 8N1 | NOR | Minicom 2.4 | VT102 | Offline
```

Figura 50

Vemos según las capturas lo que preveíamos: consumo en 2 canales y cero en los demás. El consumo que aparece en los dos canales se corresponde con la tabla del Anexo C.2, por lo que podemos confirmar que el sistema realiza una medida correcta para los valores máximos de consumo en cada uno de los dominios.

Beagle + ADC Board + Convertidor V/I

En primer lugar, vamos a comenzar con las pruebas realizadas en el convertidor para la realización de la PCB.

Lo primero que hemos realizado es el montaje de circuito en una placa de prototipado. Una vez montado el circuito, conectamos la primera parte del proyecto -Fuente de alimentación + Labview- y sacamos una señal cuadrada para verificar su comportamiento. En una primera instancia, metemos una señal cuadrada como señal de entrada de tensión 0 y 1V, de esta forma, con una relación de 1/1000, generaremos una señal cuadrada de 0 a 1mA, vemos en las Figuras 51 y 52 el resultado

En la primera imagen, nos encontramos con 0V a la salida de la fuente.

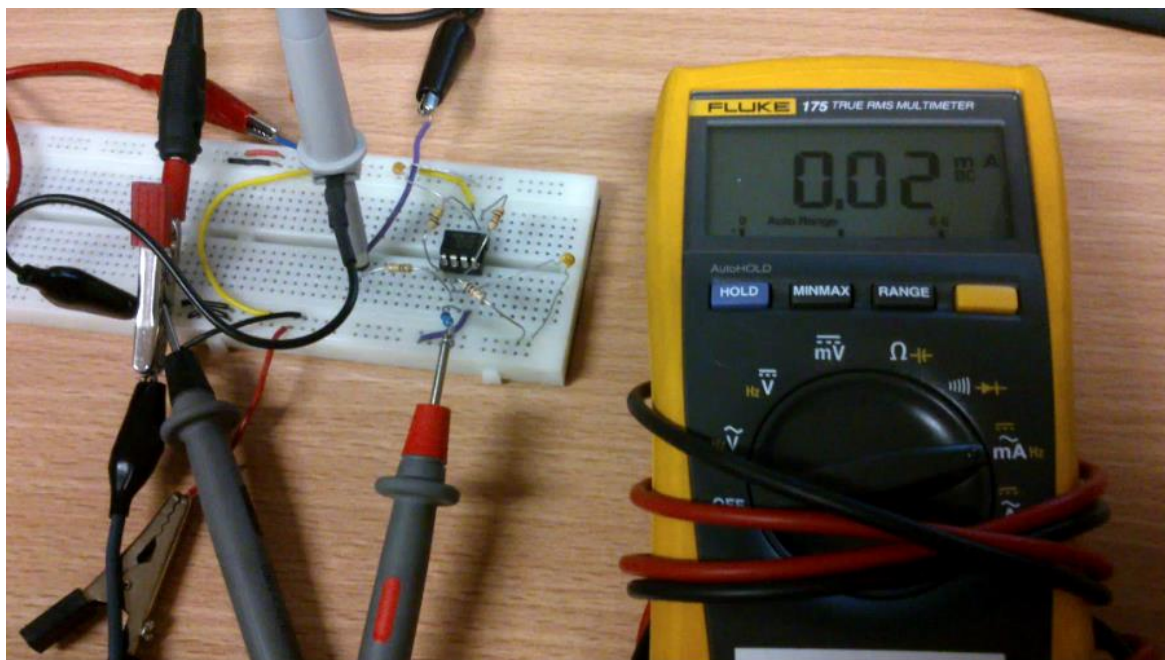


Figura 51

En esta siguiente figura, nos encontramos con 1V a la salida, lo que hará que nuestro convertidor tenga 1mA.

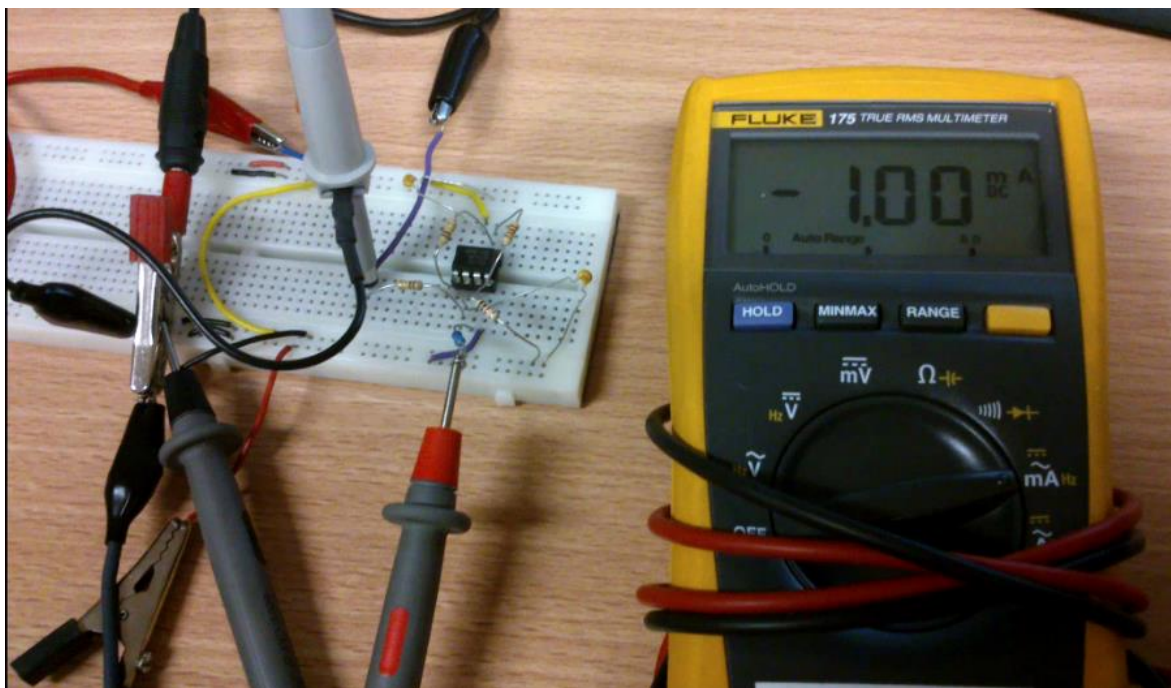


Figura 52

Una vez comprobado que somos capaces de convertir con precisión la señal de entrada, podemos proseguir con la conexión de la Parte 2 completa.

Así, conectamos el convertidor a la Fuente de alimentación con una tensión continua de 10V. Recordando los valores de la relación V/I , sabemos que la corriente será inversamente proporcional a la tensión mil veces si fijamos las resistencias calculadas. Por lo tanto, si tenemos a la entrada 10V, la Beagle tiene que darnos un valor de consumo de 10mA.

Es importante destacar que para poder llevar a cabo esta prueba, hemos necesitado de otra fuente de alimentación adicional para poder alimentar el convertidor.

Para comprobar si es estable el dato vamos a utilizar el programa compilado y modificado por nosotros mismos, donde veremos las 10 muestras recogidas y no una media de ellas:


```

root@beagleboard:/miopt# ./testi
spi mode: 0
bits per word: 8
max speed: 50000 Hz (50 KHz)
number of channels: 16
ADC automatic mode 1 has been configured

```

IO	Consumption:
	10.408051 mA 55.162670 mW
	10.536058 mA 55.841110 mW
	10.299736 mA 54.588604 mW
	10.467132 mA 55.475800 mW
	10.614834 mA 56.258621 mW
	10.792076 mA 57.198006 mW
	10.693607 mA 56.676121 mW
	10.457285 mA 55.423611 mW
	10.388357 mA 55.058296 mW
	10.240656 mA 54.275478 mW

Figura 53

En la Figura 49, hemos conectado nuestro convertidor con una tensión de entrada de 10V a nuestra Beagle Board. Esta conexión la hemos realizado al dominio de tensión que simularía el canal IO del procesador. Hemos mantenido estable el dato consiguiendo una máxima variación de 0.5 mA, lo que nos ayuda a confirmar que hemos sido capaces de mantener estable la medida.

Conclusiones

Esta parte nos ayuda a completar el sistema que se encarga del procesamiento de la señal creada en la Parte 1.

Su realización ha sido posible gracias al Proyecto ADA, comentado en numerosas ocasiones en este proyecto. Este proyecto, logró comunicar por el bus SPI una placa que contenía un ADC con una Beagle Board. La limitación de dicho sistema, es que para simular el consumo de los diferentes dominios, se emplea una resistencia, conectada en una patilla al amplificador de instrumentación y la otra a masa, para cerrar el circuito y simular los valores de tensión.

Nosotros, hemos completado dicho sistema usando un convertidor tensión-corriente que hace la función de fuente de corriente variable, la cual se encargará de cerrar sistema haciendo unas simulaciones variables en el tiempo y, por lo tanto, más reales.

Una vez conseguidas dichas pruebas, tenemos las dos partes del proyecto funcionando por separado. Para poder verificar el correcto funcionamiento del sistema completo, necesitamos montar dicho sistema, poniendo especial atención a las alimentaciones y masas. Teniendo en cuenta que actualmente estamos utilizando 2 fuentes de alimentación y 3 placas conectadas entre sí.

Si esta conexión funciona, podremos confirmar que tenemos nuestro banco de pruebas completado.

Sistema Completo

Introducción a las pruebas

Las pruebas son un resultado de la interacción entre la “Parte 1” – Labview y fuente de alimentación- y la “Parte 2” –convertidor, ADC Board y Beagle Board- de este trabajo.

Hasta ahora, veníamos viendo las comprobaciones en dos partes claramente separadas. Sin embargo, con el fin de tener una visión del sistema en todo su conjunto, hemos realizado las pruebas en un orden siguiendo el recorrido de la señal de principio a fin: desde su creación hasta su representación en la Beagle Board. Esto nos ayudará a identificar y segmentar los posibles fallos de una forma más eficiente.

Primero, vamos a conectar la primera parte del documento, compuesta por la fuente de alimentación y el ordenador. Esta vez, incluiremos el convertidor a dicha fuente para realizar una primera prueba de entrada al sistema. Con esta prueba comprobaremos si la señal que entra al convertidor es correcta y, por lo tanto, esta parte quedaría completamente comprobada.

A continuación, una vez comprobada la señal, conectaremos el sistema completo y verificaremos su funcionamiento mediante el programa modificado en C que tenemos compilado en la Beagle.

El fin de estas pruebas es, básicamente, conectar una señal que hemos generado y ver esa misma señal –ya convertida en corriente- en nuestra Beagle Board.

Primeras conexiones

Para comenzar con una prueba simple, hemos generado una señal cuadrada con la fuente de alimentación. Más tarde, una vez comprobado que la señal es correcta, la hemos conectado al convertidor para analizar con un osciloscopio la conversión.

Ya hemos visto en las conclusiones de la primer parte que somos capaces de crear una señal cuadrada de hasta 50Hz. Para realizar una primera prueba, hemos tomado una onda cuadrada de 10Hz. Dicha onda la conectamos al convertidor para más tarde mostrarla con ayuda del osciloscopio. Vemos este proceso en la Figura 55 de la página siguiente.

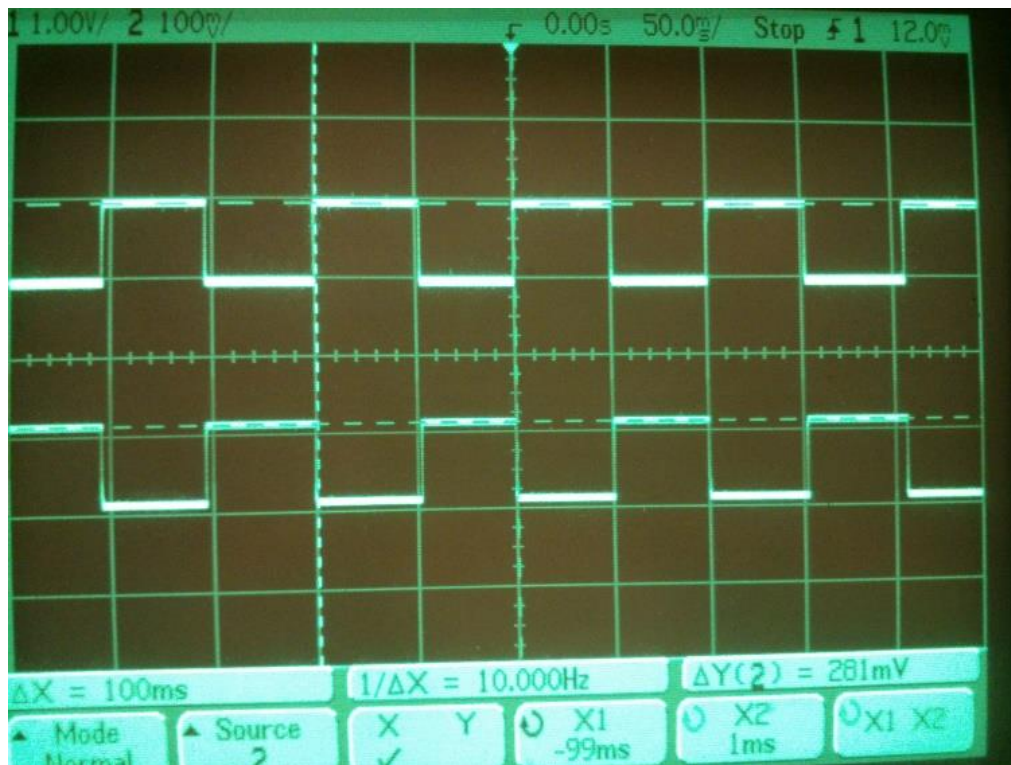


Figura 54

Como esperamos, se cumple la función que habíamos estudiado para el convertidor, donde:

$$-V_E = R_4 \cdot I_L$$

Para llevar a cabo una señal más compleja y estudiar su conversión, hacemos las pruebas con otra señal senoidal. Hemos seleccionado una señal senoidal de diferentes frecuencias para mostrar el efecto de la señal digitalizada que hablábamos en el apartado “Labview”:

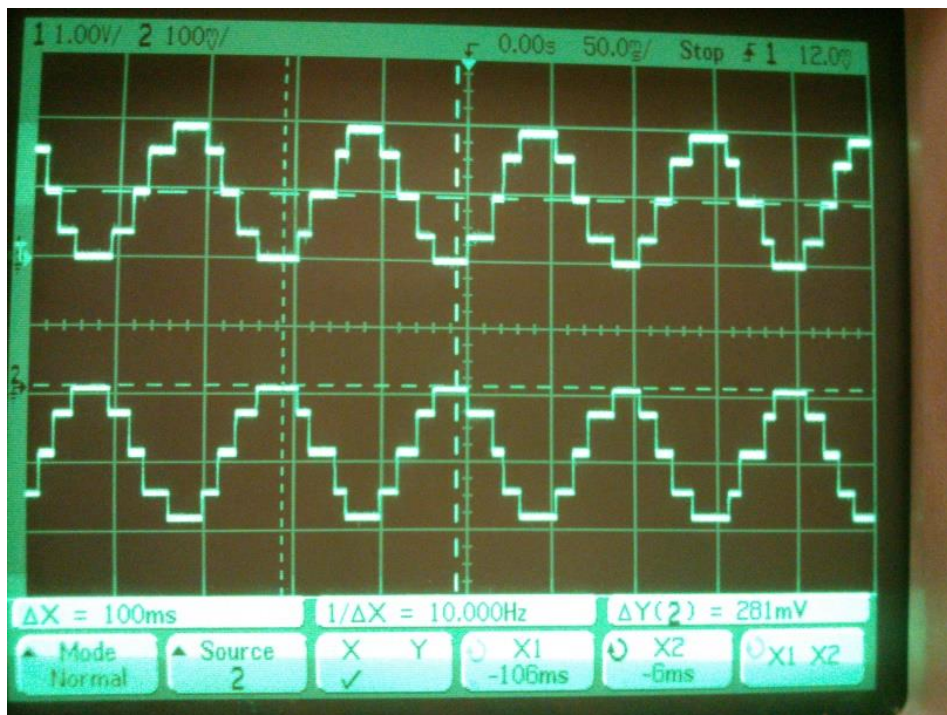


Figura 55

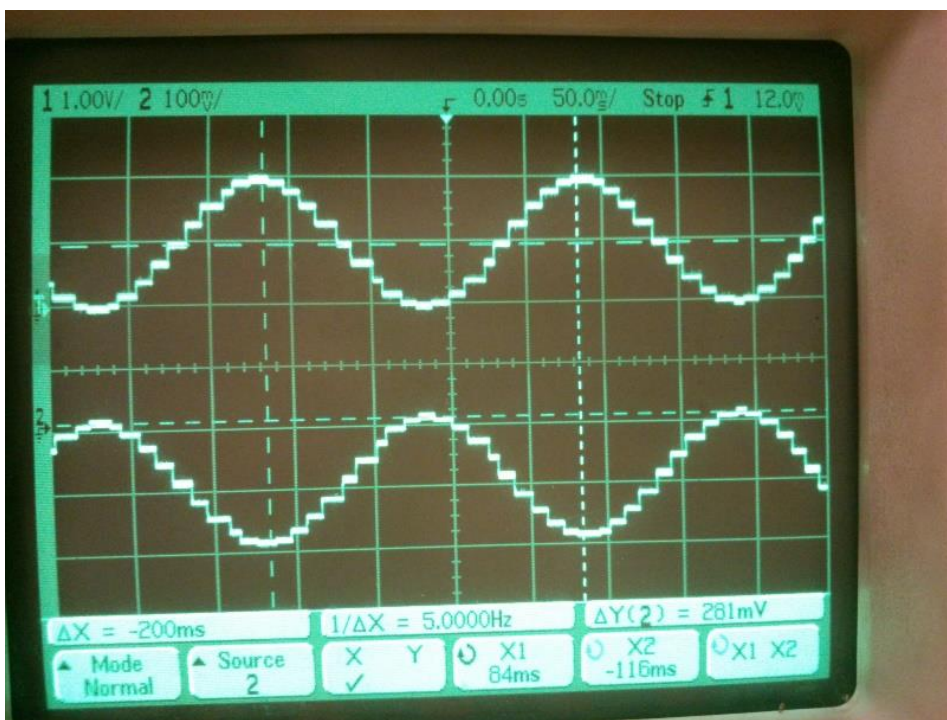


Figura 56

Podemos apreciar en los dos casos que obtenemos una señal idéntica, solamente variando la amplitud de la que hablábamos en la relación V/I . Por tanto, podemos continuar e ir al siguiente paso.

Conexión Sistema completo

Una vez obtenidas los dos tipos de señales en la salida del convertidor podemos proceder a conectarlo al apartado 2 del documento, es decir, a la ADC Board y la Beagle.

Como hemos venido viendo hasta ahora, en este paso tenemos que fijarnos bien en las conexiones del sistema por estar conectado a varias fuentes de alimentación, ya que una mala conexión podía ocasionar la pérdida de la tarjeta.

Recordamos lo que habíamos visto en el apartado del convertidor. Tenemos que elegir los valores de las Resistencias que fijarán la fórmula explicada anteriormente:

$$-V_E = R_4 \cdot I_L$$

Para nuestro caso, hemos elegido unos valores de Resistencia de $1K \Omega$, para conseguir una relación fácil de 1/000. Por lo tanto, si a la entrada colocamos una tensión de 5V, a la salida nos encontraremos con una corriente de 5 mA.

Para ayudarnos de una forma visual, mostramos en la Figura 57 una interconexión del convertidor con el sistema que vamos a conectarlo ahora mismo.

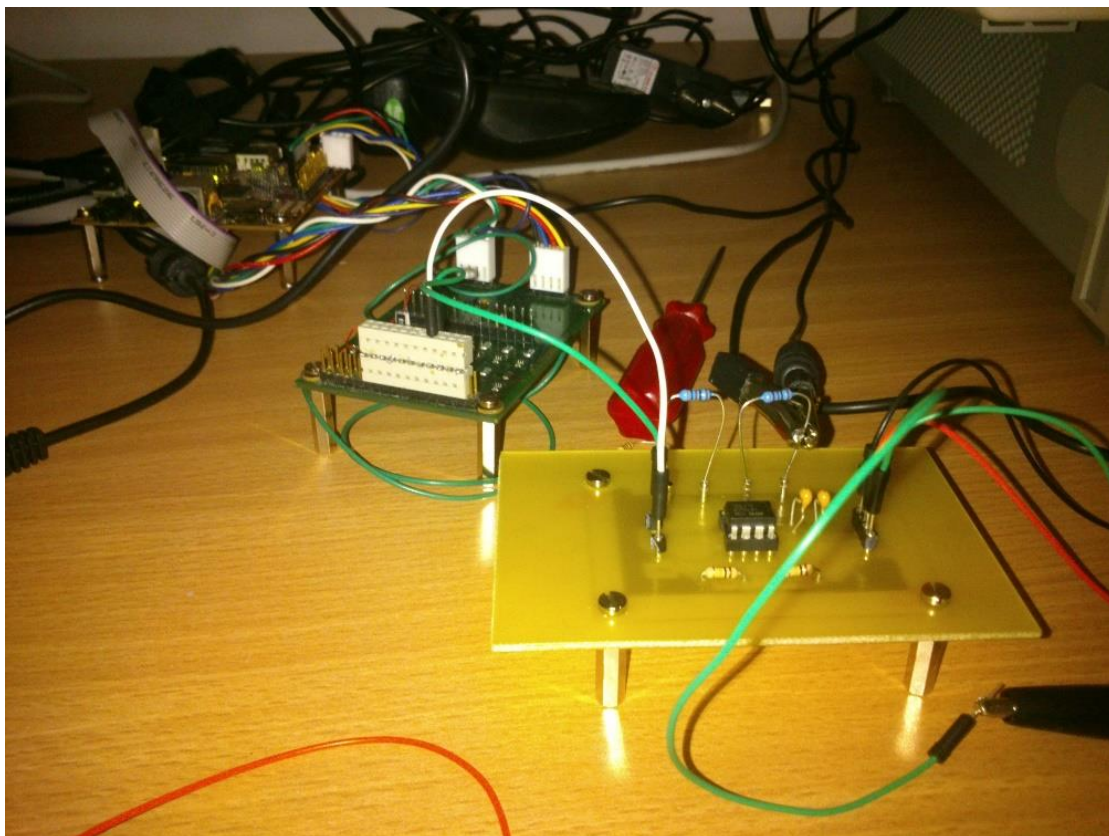


Figura 57

Como vemos en la Figura 57, nos hemos ayudado de unos jumpers para poder conectar las tarjetas entre sí. Ponemos especial atención en las entradas en las salidas, fijándonos la conexión de masa y alimentación.

Como tenemos costumbre, comenzaremos realizando la primera prueba con una onda cuadrada para acabar con ondas senoidales.

Esta vez, comenzamos dando unos valores más altos de tensión y cambiamos la relación de las resistencias para poder abarcar unos valores de consumo más elevados. Esta decisión es debida a que tenemos una fuente de alimentación que no puede sacar valores por encima de los 15 V. El valor de las resistencias será de 300Ω con lo que conseguiremos valores de 0 a 40mA –usando la fórmula del apartado “Convertidor V/I”-.

Para el caso que vamos a mostrar, tomamos valores de 5V y 10V como onda cuadrada, con lo que conseguiremos unos valores de tensión de 16mA y 33mA, respectivamente.

Vemos lo que muestra por pantalla la comunicación serie:

```
root@beagleboard:/miopt# ./testi
spi mode: 0
bits per word: 8
max speed: 50000 Hz (50 KHz)
number of channels: 16
ADC automatic mode 1 has been configured

IO      Consumption:
..... 16.375202 mA 86.788574 mW
        16.995548 mA 90.076408 mW
        16.503210 mA 87.467018 mW
        16.759228 mA 88.823914 mW
        16.660757 mA 88.302017 mW
        33.744926 mA 178.848114 mW
        33.744926 mA 178.848114 mW
        33.823704 mA 179.265640 mW
        33.754776 mA 178.900314 mW
        33.607075 mA 178.117508 mW
```

Figura 58

Una vez verificados los datos, vamos a proceder a la creación de una onda seno y veremos si la comunicación serie es capaz de mostrarla sin saltarse ninguna muestra.

También vamos a realizar la prueba en varios canales para verificar que todos funcionan correctamente.

```
root@beagleboard:/miopt# ./testi
spi mode: 0
bits per word: 8
max speed: 50000 Hz (50 KHz)
number of channels: 16
ADC automatic mode 1 has been configured

IO      Consumption:
0.000000 mA 0.000000 mW
0.000000 mA 0.000000 mW
0.000000 mA 0.000000 mW
0.000000 mA 0.000000 mW
0.000000 mA 0.000000 mW
0.000000 mA 0.000000 mW
0.000000 mA 0.000000 mW
0.000000 mA 0.000000 mW
0.000000 mA 0.000000 mW
0.000000 mA 0.000000 mW
0.000000 mA 0.000000 mW
0.000000 mA 0.000000 mW

SRAM    Consumption:
0.068927 mA 0.365316 mW
0.472646 mA 2.505022 mW
1.723187 mA 9.132893 mW
2.008744 mA 10.646343 mW
1.496711 mA 7.932570 mW
0.315097 mA 1.670015 mW
0.029540 mA 0.156564 mW
0.974832 mA 5.166608 mW
1.664106 mA 8.819764 mW
2.176139 mA 11.533538 mW
```

Figura 59

Apreciamos en este canal una onda senoidal de amplitud 2V. Y observamos cómo nos encontramos con un consumo nulo en el canal IO.

Con estas capturas podemos dar por completo el apartado de pruebas.

Conclusiones Sistema Completo

Conclusiones

Una vez conocidas todas las partes y explicado su funcionamiento, vamos a proceder al cierre del documento llegando a las conclusiones que podemos recoger del trabajo realizado a lo largo del proyecto.

En un principio, el objetivo propuesto era el desarrollo de un sistema en Labview que nos permitiese controlar el consumo de corrientes en las diferentes aplicaciones que puede suministrar un sistema empotrado.

Sin embargo, la fuente de alimentación sólo dispone de una salida y, por lo tanto, de una vía de comunicación con el Labview. Esta limitación hace que el objetivo primeramente propuesto deje una línea abierta para otros proyectos.

Por el motivo previamente citado, y una vez estudiada la limitación, decidimos cambiar el rumbo del proyecto fijando el objetivo en un sistema capaz de generar una señal que se corresponda con uno de los canales de consumo de un sistema empotrado. Marcado este nuevo objetivo, podemos afirmar que los resultados obtenidos han sido satisfactorios.

Somos capaces de simular señales de hasta 50Hz y somos capaces de tratar dicha señal para que sea procesada en un sistema empotrado. Así pues, hemos desarrollado un sistema automático de medida que genera una forma de onda la cual simula una de las corrientes que sirven como entrada para verificar el sistema de adquisición el cual, junto con la tarjeta Beagle Board, permitirá la toma de decisiones en relación con el consumo de energía.

Posibles Mejoras

En el ámbito de nuestro proyecto, una de las mejoras más significativas sería realizar una interfaz capaz de sacar a partir de una salida de la fuente de alimentación 16 canales para realizar la simulación completa de nuestro sistema. Sería interesante poder realizar esta operación para una mayor interacción con la Beagle Board, pero dada la complejidad del apartado, podría ser una mejora significativa.

Otra de las posibles mejoras sería separar la acción humana de elegir los datos y llevarlo a algún tipo de automatizado, para no tener que intervenir cada vez que queramos realizar algún cambio en la señal.

También podríamos dotar al programa de un apartado gráfico en el que realizar las simulaciones, e incluso poder dibujar las señales que queramos sacar por la fuente de alimentación, sin tener que definir el tipo. Con esto conseguiremos poder dibujar unas señales más reales y de acorde al consumo del sistema empotrado.

Más mejoras en cuanto al apartado hardware, se ha realizado un proyecto en el cual se ha unido la ADC Board y la Beagle Board, por lo tanto sería interesante seguir esa línea de trabajo para poder seguir desarrollando esta línea de investigación de consumo, cada vez más importante en la actualidad.

Bibliografía

[1] *USER'S GUIDE* - Agilent Technologies Model 66319B/D, 66321B/D Mobile Communications DC Source

<http://literature.cdn.keysight.com/litweb/pdf/5964-8184.pdf>

[2] *Quick Start Guide* - Agilent Technologies 14565B Device Characterization Software

<http://literature.cdn.keysight.com/litweb/pdf/5969-2943.pdf>

[3] NI LabVIEW - Wikipedia

<http://es.wikipedia.org/wiki/LabVIEW>

[4] Getting Started with LabVIEW

<http://www.ni.com/pdf/manuals/373427j.pdf>

[5] Introducción a Labview – *Mariano Ruiz González y Guillermo de Arcas Castro – Madrid 2001*

[6] Menu de ayuda de Labview – Driver Agilent

[7] HAMBLEY, A.R. ELECTRONICA – *Allan R. Hambley* - 2001

[8] HAMBLEY, A.R. ELECTRONICA – *Allan R. Hambley* – 2001. – Capitulo 2 Amplificadores Operacionales

[9] Capitulo 7 – Acondicionamiento de señales

http://www.ctr.unican.es/asignaturas/instrumentacion_5_IT/IEC_7.pdf

[10] Manual Kicad

http://sluc.org.ar/sites/default/files/KiCad_tutorial.pdf

[11] Practica 1 – Asignatura de circuitos electrónicos. Profesor Manuel Vazquez. Curso 2º.

[12] Peripheral SPI use with OMAP3530 – Advanced digital Architectures. *Proyecto de Miguel Chavarrias*. 2012.

[13] SPI

<http://serials3.software.informer.com/wiki/index.php?title=Programs:Serial>

[14] Wiki Beagle Board

<http://en.wikipedia.org/wiki/BeagleBoard>

[15] Datasheet Beagle

http://beagleboard.org/static/BBSRM_latest.pdf

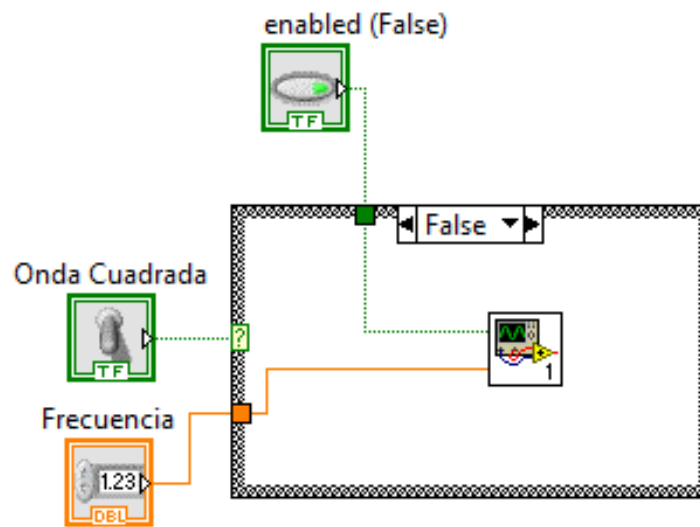
[16] Intranet GDEM

<https://blog.gdem.sec.upm.es>

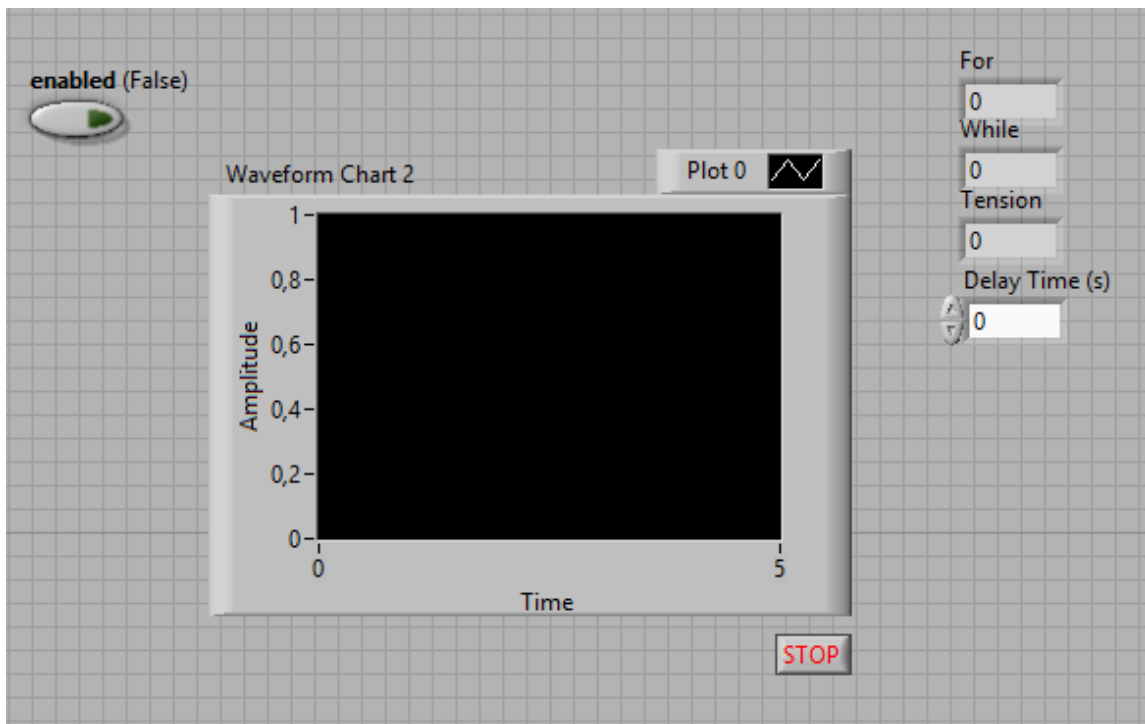
Otros datos de interés:

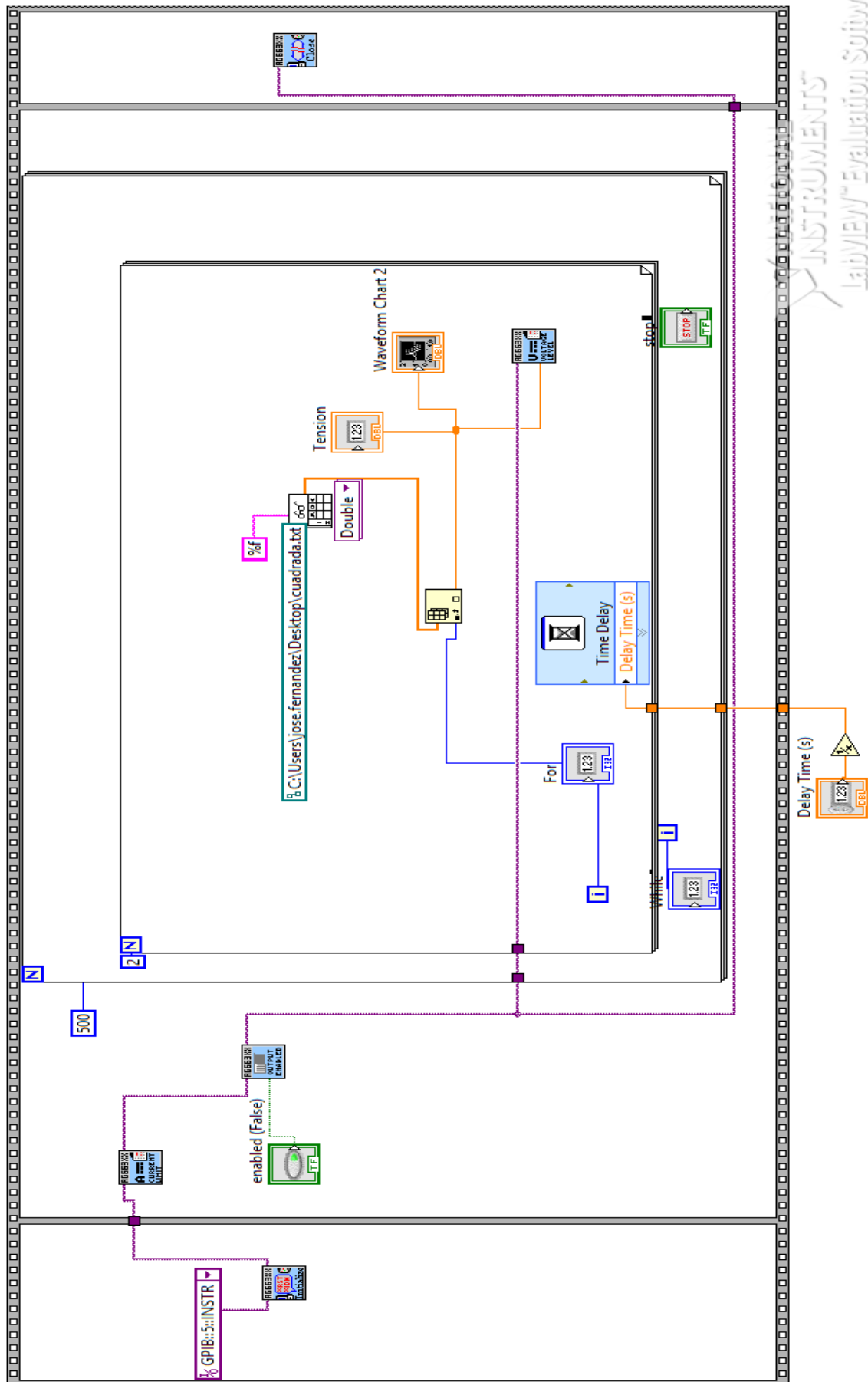
- Consulta de artículos:

- o ¿Qué es un sistema embebido? - <http://electronica-embebida.blogspot.com.es/2011/06/que-es-un-sistema-embebido-o-como.html>

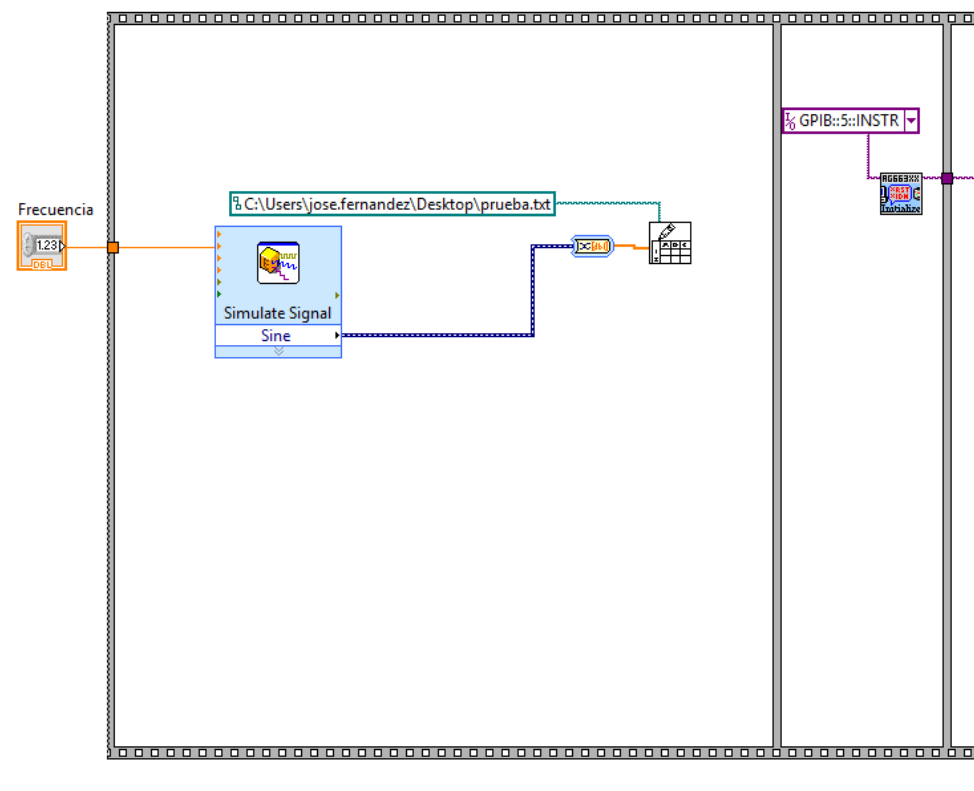
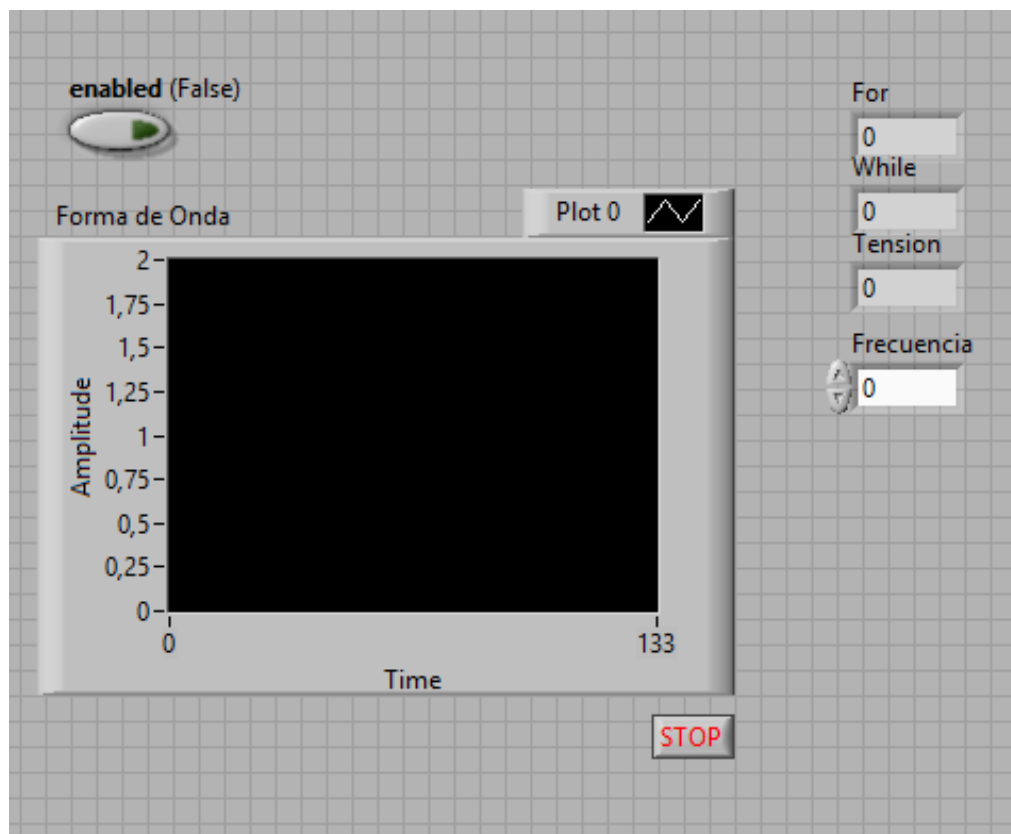


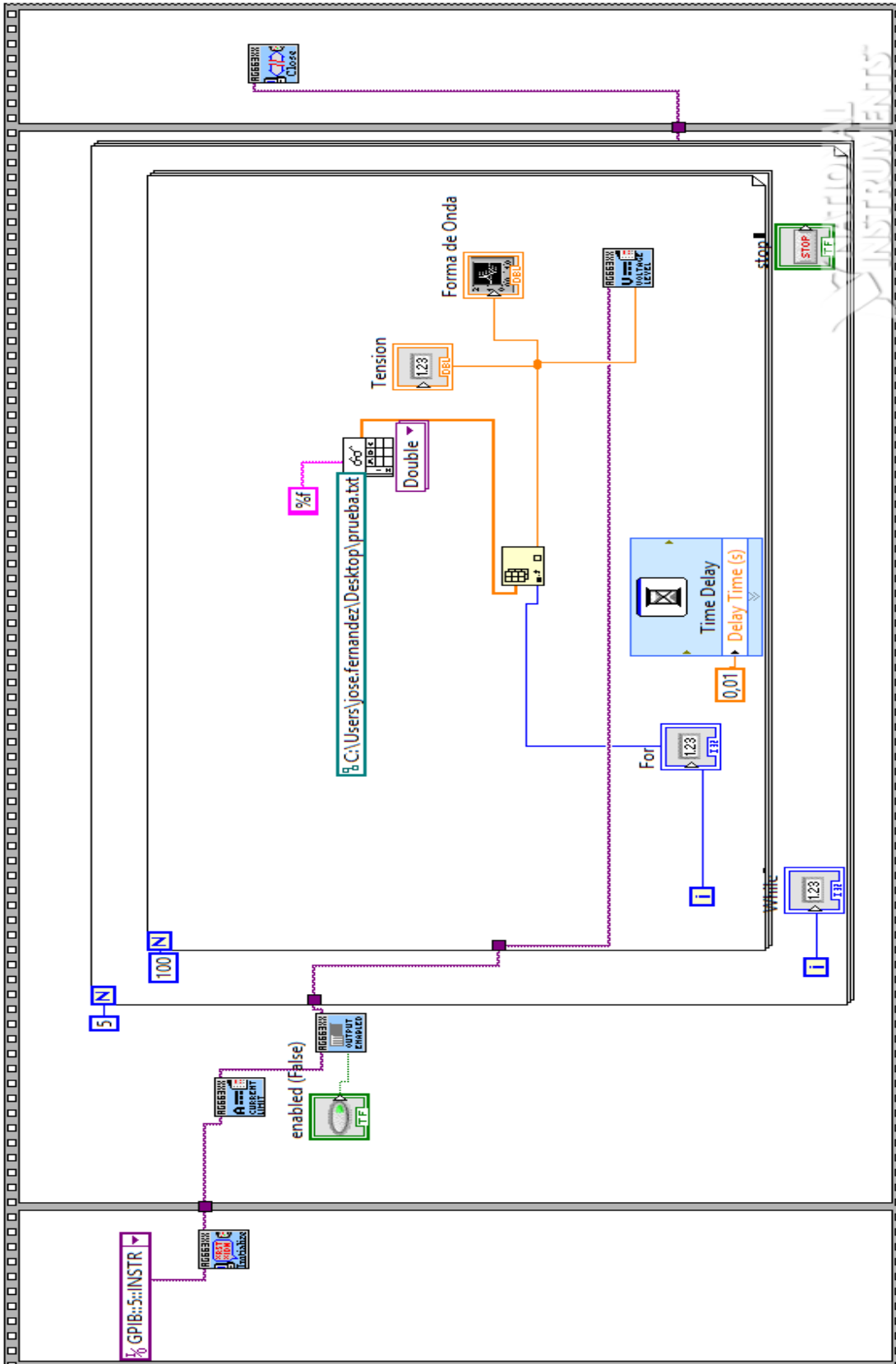
Función Cuadrada





Función Seno



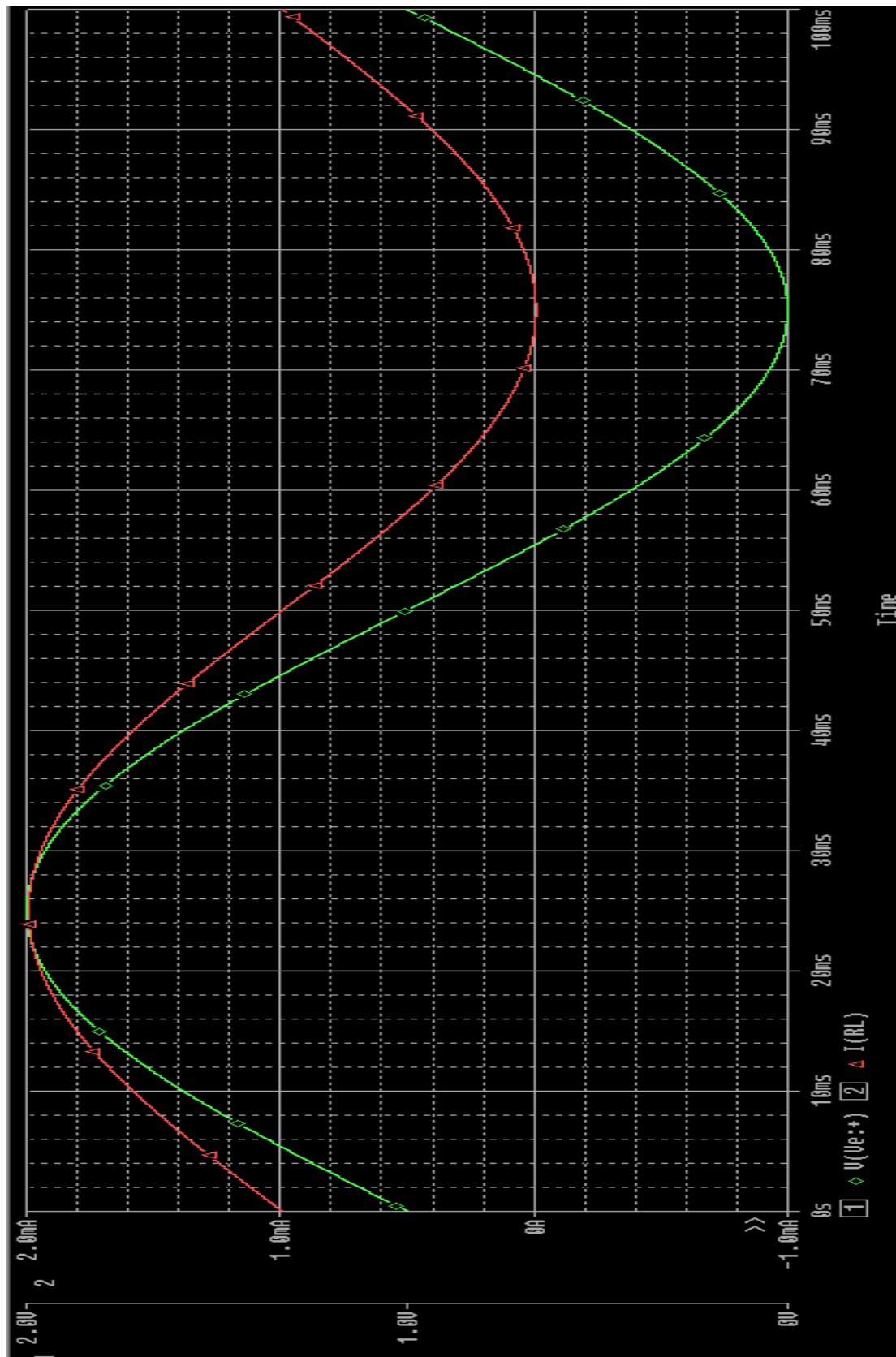


Anexo A.2 – Representación Onda Seno

X	SEN 10Hz	SEN 20Hz	SEN 30Hz
0	0	0	0
0,005	0,309016994	0,58778525	0,80901699
0,01	0,587785252	0,95105652	0,95105652
0,015	0,809016994	0,95105652	0,30901699
0,02	0,951056516	0,58778525	-0,58778525
0,025	1	0	-1
0,03	0,951056516	-0,58778525	-0,58778525
0,035	0,809016994	-0,95105652	0,30901699
0,04	0,587785252	-0,95105652	
0,045	0,309016994	-0,58778525	
0,05	1,22515E-16	0	
0,055	-0,30901699		
0,06	-0,58778525		
0,065	-0,80901699		
0,07	-0,95105652		
0,075	-1		
0,08	-0,95105652		
0,085	-0,80901699		
0,09	-0,58778525		
0,095	-0,30901699		
0,1	0		

Anexo B – Convrtidor V/I

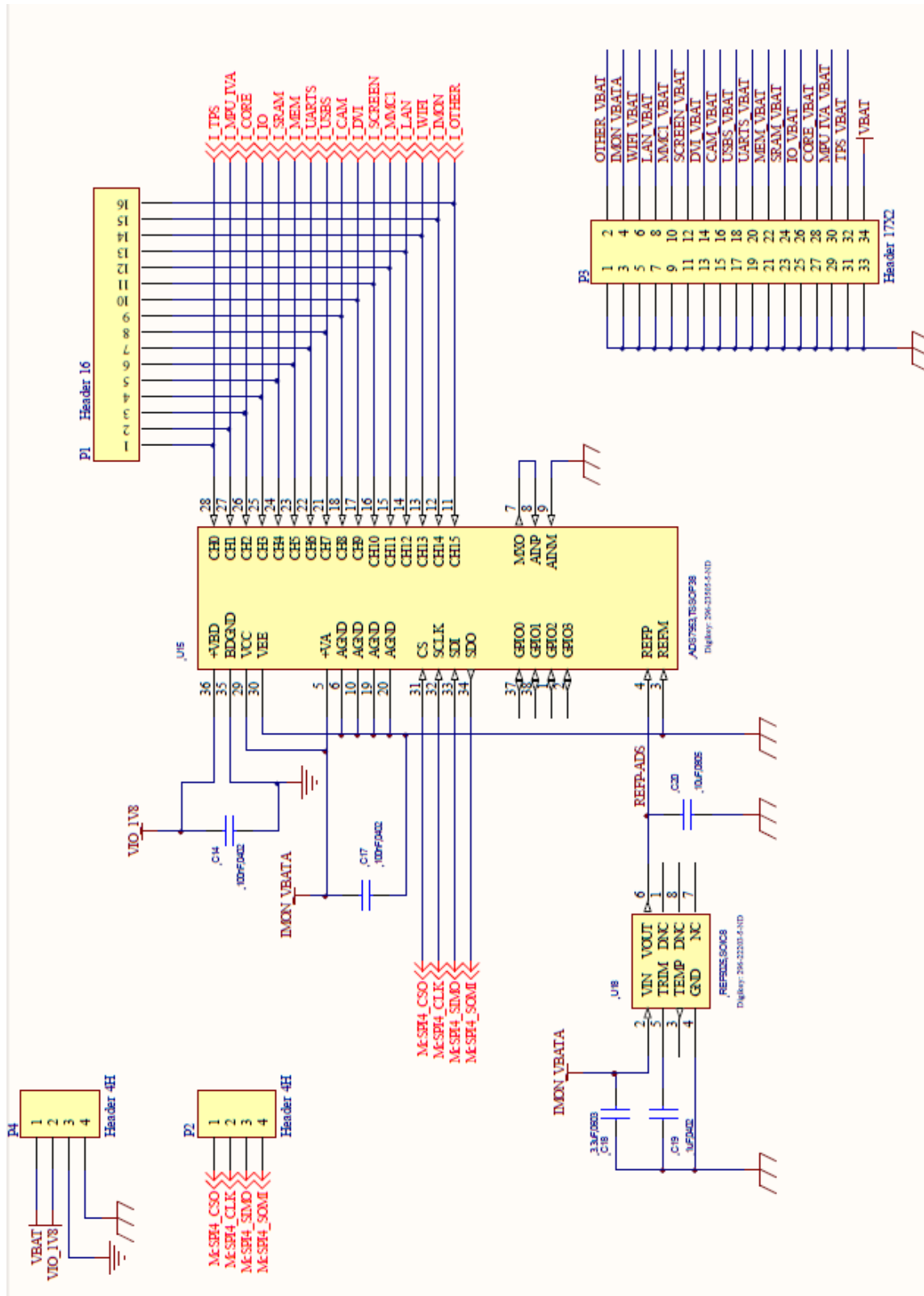
Simulación PSpice



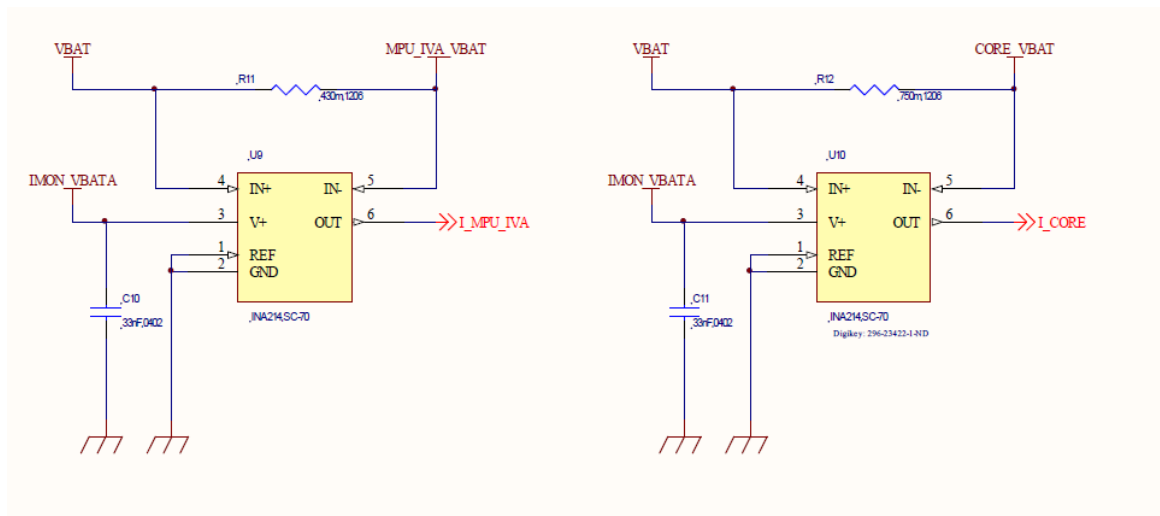
Anexo C – ADC Board

Anexo C.1 – Esquemático ADC Board

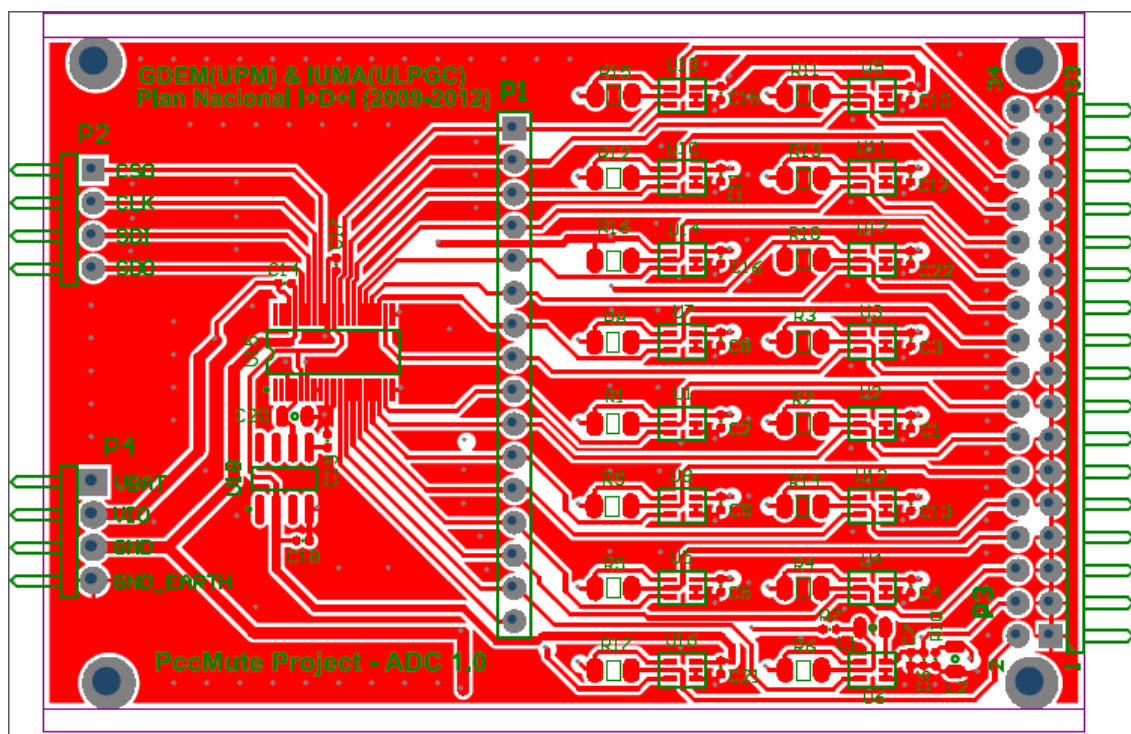
- ADC



- Canales



- Huella



ADC channel	ADC pin	conn_P1	conn_P3	Voltage domain	Rsense value	Resistor identifier	Amplifier identifier	Full scale	Max. Current	Sense resistor* value	Recommended resistor* value	Max. power of sense modules W
CH0	29	1	32	TPS	0,47	R15	U13	0,025	0,05	94	120	0,265957447
								0,025	0,06	86	100	0,290697674
CH2	26	3	28	CORE	0,45	R12	U10	0,025	0,06	90	100	0,277777778
								0,025	0,04	124	150	0,201612903
CH4	24	5	24	SRAM	0,56	R16	U14	0,025	0,04	112	150	0,080357143
CH5	23	6	22	MEM	0,62	R18	U17	0,025	0,04	124	470	0,072580645
CH6	22	7	20	UARTS	0,1	R8	U7	0,025	0,25	20	27	1,25
CH7	21	8	18	USBS	0,033	R3	U3	0,025	0,76	6,6	56	3,787878788
CH8	18	9	16	CAM	0,12	R1	U1	0,025	0,21	24	27	1,041666667
CH9	17	10	14	DVI	0,068	R2	U2	0,025	0,37	13,6	18	1,838235294
CH10	16	11	12	SCREEN	0,068	R9	U8	0,025	0,37	13,6	18	1,838235294
CH11	15	12	10	MMC1	0,11	R14	U12	0,025	0,23	22	27	1,136363636
CH12	14	13	8	LAN	0,2	R5	U5	0,025	0,13	40	47	0,625
CH13	13	14	6	WIFI	0,068	R4	U4	0,025	0,37	13,6	27	1,838235294
CH14	12	15	4	IMON	0,11	R10	U6	0,025	0,23	22	27	1,136363636
CH15	11	16	2	OTHER	0,91	R17	U16	0,025	0,03	182	220	0,137362637